



A CAN Kingdom

Rev. 3.01

by

Lars-Berno Fredriksson

1990,91,92,95, 960101 ©KVASER AB

Abstract.

The CAN protocol has a great hidden power that is not obvious until you have penetrated the problems around designing distributed embedded controls systems where independently developed modules will be integrated into the system. CAN Kingdom is especially developed for such systems. Most networks provide services for connected modules. In *control networks* however, connected modules serve the network. CAN Kingdom takes advantage of this fact, making separation of module and system development not only possible but even simple.

In fact, most papers written on the subject of Controller Area Network design are, in my opinion, misleading as they are built upon the OSI model. The OSI model was developed for communication networks. Such networks are intended to be accessed by users, unknown at the design phase, during runtime. In a controller area network, where real-time features are essential, all communication needs have to be known at the design phase. CAN Kingdom is open only during this phase. At runtime a CAN Kingdom network works exactly in the way the network designer has decided.

CAN Kingdom unleashes the full power of CAN. It gives a **system designer** maximum freedom to create his own system. He is not bound to the CSMA/AMP multi-master protocol of CAN but can create systems using virtually any type of bus management and topology. CAN Kingdom opens the possibility for a **module designer** to design general modules without knowing in which system they will finally be integrated into and what type of higher layer CAN protocol it will have. As the system designer can allow only specific modules to be used in his system, the cost advantage of an open system can be combined with the security of a proprietary system!

Since the identifier in a CAN message not only identifies the message but also governs the bus access, a key factor is the enumeration of the messages. Another important factor is to see to it that the data structure in the data field is the same in both the transmitting and receiving modules. By adopting a few simple design rules these factors can be fully controlled and communication optimized for any system. This is done during a short setup phase at the initialization of the system. Including some modules not following the rules of the CAN Kingdom into a CAN Kingdom system is even possible.

CAN Kingdom also enforces a conform documentation of modules and systems.



BLANK PAGE

TABLE of Contents

Preface to Rev 3.01	V
Preface to Rev 3.0	V
1. Introduction.	1
2. Some CAN Kingdom basics.	4
2.1 Who can use CAN Kingdom?	6
2.2 Some characteristics of CAN Kingdom:	6
3. CAN Kingdom symbols.	7
3.1 Hardware symbols.	8
3.1.1 Some remarks about the Postal System and Post Offices.	8
3.2 Information symbols.	9
3.3 Parameters that have to be defined before a CAN Kingdom setup phase takes place.	10
3.4 The setup phase in a simple CAN Kingdom.	10
3.5 The Letter distribution.	12
4. CAN Kingdom building blocks.	13
4.1 The Form.	13
4.2 The Document.	15
4.3 Lists.	16
4.4 The Folder.	17
4.5 The Envelope.	19
4.6 Compressed Documents and Letters.	20
4.6.1 Compressed Pages in regular Envelopes.	21
4.6.2 Regular Pages in Compressed Envelopes.	28

4.6.3	Fully Compressed Letters.	32
4.7	Filters.	32
4.7.1	Filters for extracting First Identifiers.	32
4.7.2	Hardware filters.	37
4.8	The use of Documents, Folders and Lists. A simple example.	39
6.	Bus Management.	49
6.1	Event driven types.	49
6.2	Daisy chain types.	51
6.3	Time driven types.	54
7.	The Block.	55
7.1	The Block Transfer Document.	55
7.2	Block Transfer Procedure.	56
8.	CAN Kingdom Documents.	58
8.1	The King's Document.	58
8.1.1	The King's Page 0 . (Compulsory) Terminates the setup phase and sets a City in appropriate modes.	59
8.1.2	The King's Page 1 . (Compulsory) Initiating Page. Provides the Base Number and asks for Mayor's response.	61
8.1.3	The King's Page 2 . (Compulsory) Assigning an Envelope to or expelling an Envelope from a Folder forming Letters and restraining the use of an Envelope.	62
8.1.4	The King's Page 3 . (Optional) Assigning a City to Groups.	65
8.1.5	The King's Page 4 . (Optional) Removing a City from Groups.	66
8.1.6	The King's Page 5 . (Optional) Action/Reaction Setup Page.	67
8.1.7	The King's Page 8 . (Optional) Postmaster control register settings.	68
8.1.8	The King's Page 9 . (Optional) Change of City physical address.	70

8.1.9 The **King's Page 10**. (Optional) Minimum time elapsing between two consecutive transmissions of the same Envelope. 71

8.1.10 The **King's Page 11**. (Optional) Circular Time Base Setup Page. 73

8.1.11 The **King's Page 12**. (Optional) Repetition Rate and Open Window Setup Page. 74

8.1.12 The **King's Page 16**. Setting the Folder Label and/or placing a Document into a Folder. Placing a Document into a Folder. 75

8.1.13 The **King's Page 17**. Create a Document, a Form or a Line from predefined Form-, Line- or Bit-Lists. 77

8.1.14 The **King's Page 18**. Create a Compressed Form from Presentation Lists. 79

8.1.15 The **King's Page 19**. First Identifier mask 81

8.1.16 The **King's Page 20**. Create filters for a Postmaster. 82

8.2 The Mayor's Document. 84

8.2.1 The Mayor's Pages 0 and 1 for City Identification. 84

8.2.2 The Envelope assigned to the Folder 1, the Mayor's Document Folder. 87

8.2.3 The Mayor's Document Folder Label. 88

8.3 Block Transfer Document. 89

8.4 Time Herald Document. 95

8.5 Linear Time Document. 96

9. Start-up procedure. 97

10. CAN Kingdom Product Identification Rules. 98

11. Graphical presentation of a CAN Kingdom. 100

12. CAN Kingdom Glossary. 103

13. References. 105

Appendix 106



BLANK PAGE

Preface to Rev 3.01

Compared with revision 3.0 mainly editorial changes have been made into revision 3.01. Misprints have been corrected, some figures enhanced or added and some clarifying text has been added. In former revisions transmit has been denoted x.1 and receive x.0 in Lists. This is now changed into prefix T and R respectively, corresponding to an established common praxis. A new paragraph has been added, 4.7 Filters, to clarify how filters are used for Compressed Envelopes and for filtering out specific Envelopes. An appendix has been added clarifying the use of Intel and Motorola byte definitions in Compressed Documents.

I would like to thank Daniel Berglund for his contributions to the Filter paragraph and the appendix.

Kinnahult 960101
Lars-Berno Fredriksson

Preface to Rev 3.0

CAN Kingdom¹ Rev. 3 is a substantial revision and extension of CAN Kingdom Rev. 2.2. The reader familiar with Rev. 2.2 will find the basics still the same but that some King's Pages are reworked and/or renumbered. He will also find new items as Compressed Documents and Letters, Kingdom Time, the Mayor's Document as well as new features like Block Transfer and Action/Reaction Folders. King's Pages for construction of software and hardware filters are introduced. Standardized procedures for identification of Cities and startup of a system are defined, ensuring a high level of safety. The King's Envelope does not have to be zero any longer but can be set by the Kingdom Founder. By Revision 3 CAN Kingdom is now much more powerful and versatile but at a cost: Revision 2.2 and 3 are not fully compatible.

By CAN Kingdom a multitude of different serial protocols based on the low level Controller Area Network (CAN)² protocol can be built up. CAN Kingdom is an open protocol and there is no license fee for CAN Kingdom as such. However, KVASER holds a number of patents in the area of distributed embedded control systems which can be applied on systems following CAN Kingdom as well as other protocols. KVASER is prepared to give licenses on fair and reasonable terms. The use of CAN Kingdom, its applicability and suitability, for any practical use, including any patent infringements arising therefrom, is the sole responsibility of the user.

I would like to thank Kent Lennartsson, Klas Sehlstedt and Daniel Berglund, all colleagues at KVASER, for their valuable contributions to and implementations of CAN Kingdom. I would also like to thank members of CAN Textile Users Group and CAN Hydraulic Users Group who have been patient discussion partners and shared experiences of practical use of CAN Kingdom.

Kinnahult 950401
Lars-Berno Fredriksson

¹ CAN Kingdom is a registered trade mark of KVASER AB.

² The Controller Area Network (CAN) protocol is the intellectual property of Robert Bosch GmbH.



BLANK PAGE

1. Introduction.

CAN takes care of the low levels of a communication protocol but this is not enough for a network communication. Such always requires higher protocol layers. Most CAN systems running today use proprietary higher layer protocols. Often a system designer would like to use standard units, complete with mechanics and built in node electronics, like robot actuators, "intelligent" electric motors, valves, etc. In order to use such devices we will need a fully open protocol including higher protocol layers. One way to achieve this is to specify any message to be used. The SAE protocol J1939 for Bus & Trucks is an example of this approach. The fact that a CAN message identifier does not only identify the message but also gives it a certain priority on the bus makes the J1939 approach a difficult task for a system designer: The priority of a message is set by the protocol and this priority may be inappropriate in some systems, especially real-time control systems. The freedom of system designers is very much restricted. Another disadvantage is that the defined messages will only belong to a certain market, in this case the bus & trucks market. A certain device, e.g., an electric motor, may be suitable for use in different markets and has then to be produced in several versions, differing only in the protocol implemented. There are other higher layer protocols, not as rigid as J1939, e.g., DeviceNet, Smart Distributed Systems and CAN Open. However, they all need "profiles" for making different types of modules compatible in systems. By the profiles we are back into rigidity but in a more confusing way as different profiles may be contradictive to each other.

CAN Kingdom is a solution to overcome these problems. It does not present a complete higher layer protocol but a set of protocol primitives. The software of each module contains a subset of these protocol primitives which a system designer can use to build up his own final protocol during a start-up phase of his system.

CAN Kingdom is developed on a most important basic principle:

The Modules are to Serve the Network (MSN).

This implies that any module connected to a network only has to announce that it is connected. Then one or more nodes will provide this new node with the information it will need to fulfill its task in the network. *The usual concept of communication networks is the other way round: **The Network Serves the Modules (NSM)***. Then, when a module is connected to a network it already knows its role in the system and asks for the means it will need. Understanding the difference between the MSN and NSM principles is important. If you design a system based on the **MSN** principle, the network will be **tailor-made to the needs** of your machine system. This is possible as all **system requirements are known a priori**. If you design your machine on the **NSM** principle, the **qualities of the network are given a priori** and you have to design your machine according to the requirements of this specific network protocol. Any office network is built according to the NSM principle. It has to be as, at most nodes, human beings will use the network in an unpredictable way (and with low real-time requirements). This is not so in a machine network where most nodes will act in a deterministic way (with high realtime requirements). This fact makes it possible to put the usual concept of a network design upside-down and work according to the MSN principle. You will find this making many problems in machine control systems easier to solve!

A consequence of the MSN principle is that one node in the network has all information needed to initiate the system. The system designer is responsible for the software implemented in this node and he

can then decide which, and on what conditions, nodes will be accepted by the system. An advantage of this approach is that an end-user, e.g., a robot owner, cannot ruin the real-time performance of the robot controller network by connecting any transmitting CAN module, not foreseen by the system designer. Thus the system designer can take full responsibility for his system.

CAN Kingdom is developed specifically for machine control system use. My own first experience in the technology of distributed embedded control systems was an industrial robot project (ref. [1]). During the work on this project I found, as project manager, that we needed a way to describe the functionality of the system that could be understood by all participants, regardless if their background was in mechanics, electronics, feedback control, chemistry, robotics, maintenance, spray-up technique, etc. This was required in order to work out and check a proper system specification. I mention this as I frequently get remarks on why not using a language in CAN Kingdom that is familiar to specialists. As every specialty has a mumbo jumbo of its own, pleasing everyone is difficult. Maybe the language used in CAN Kingdom does not satisfy anyone at first but it allows everyone to be involved in the system specification. It is important that not only computer and communication specialists but designers, maintenance people, marketing people, etc. and even users of the system, are really participating in the specification work.

In CAN Kingdom the system is described as a country, a Kingdom, with a Capital and Cities (**Figure 1**). The King in the Capital rules the Kingdom and each City has a Mayor responsible for the local government. The only way to communicate within the Kingdom is by mail. The CAN network is described as the royal Postal System and each City has a Post Office and a Postmaster, a symbol for a CAN Controller.

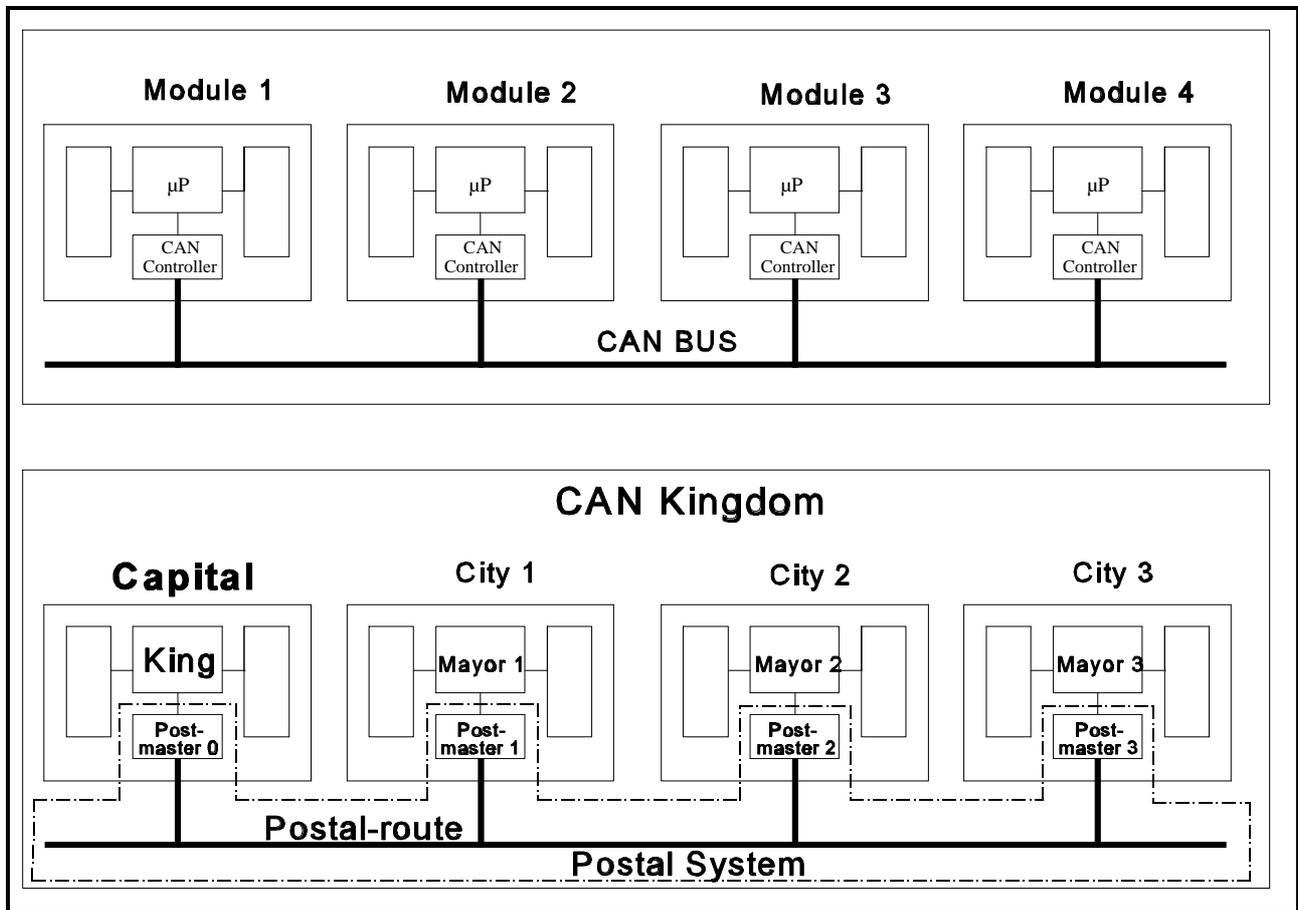


Figure 1. Conventional and CAN Kingdom representation of a CAN network.

Each City produces something and can import or export information by mail. (E.g., an actuator City produces movements and measurements. Measured values can be exported to other Cities and set point values are imported from another City or Cities). The Mayor of a City has organized any export or import information into lists. These lists are a part of the module documentation. As a system designer has chosen specific modules to be used in his machine, he also has full knowledge about these lists, i.e., a local identifier for each variable. By assigning CAN identifiers to these variables he can create an optimized protocol for his machine network. To do this he designs one module, symbolized by the Capital and the King. The simile lends itself to be extended into the Cities (nodes). You may talk about factories, offices, etc. and their management. Then you might apply concepts and theories found in the science field of human organizations in your design work (ref. [2]).

It is essential that everyone involved in a CAN project have a basic understanding of not only how the machine will work but also about fundamental problems of real-time parallel computing. Real-time parallel computing is a cornerstone in machines controlled by a network and most often only one or two persons in a machine development project are familiar with this subject. The CAN Kingdom simile visualizes how the different modules interact and highlights the timing problem. Anyone knows that it takes some time to write a letter and mail it, some time for the delivery, some time to read it and some time to act upon the contents of the letter. CAN Kingdom is not only a way to specify a communication protocol but also a framework to specify and verify a machine and its building blocks. By a specification according to CAN Kingdom you also have a framework for your project plan. This is a result of using the principle "Modules Serve the Network."

A great advantage with the CAN Kingdom approach of not being a complete protocol but a set of protocol primitives is, especially for real-time systems, that the system designer can choose a, for his application, suitable topology and bus access management. Although CAN is specified as a CSMA/AMP multi-master protocol, the final protocol can be of another type, e.g., a logical slotted ring or a token bus, depending on the opinion of the system designer and the protocol primitives provided in the nodes. With the primitives in CAN Kingdom one node may be a master controlling the network by polling the other nodes as in a traditional master-slave system like the Bitbus protocol or the FIP protocol. Making a daisy chain type or token bus type protocol with the current CAN Kingdom version is also possible. A method to resolve queues in event driven systems in a way that can give any message a predictable maximum latency time is provided. CAN Kingdom has also got a tool for setting up a global clock and creating true time driven and synchronous protocols are then possible. If CAN is used in a CA (Collision Avoiding)¹ type of protocol, the inherent arbitration procedure will increase the robustness of the net. Even if one module loses its synchronization, the network will not go down completely and a well-designed network will be capable of recovery or a graceful degradation. Even in a CA protocol based on CAN, alarm messages can be sent immediately in a safe way as collisions are resolved by arbitration.

Another advantage of CAN Kingdom is that it does not require CAN identifiers to be used as true message identifiers for the applications. Sometimes extending the identification field into the data field or placing data into the CAN identifier field is advantageous. In fact, it is a pity that the arbitration field of a CAN message is denoted "Identifier field" as this may preclude many an ingenious idea on an efficient protocol construction.

¹ CAN is often erroneously called a CA type of protocol . Messages do collide in CAN but collisions are resolved in a non-destructive way by the bit-wise arbitration. Examples on Collision Avoidance types are synchronous protocols, token protocols and TDMA (Time Division Multiple Access) protocols.

2. Some CAN Kingdom basics.

The CAN Kingdom is a set of protocol primitives, based upon the CAN protocol from Bosch, and a tool for a system designer to create an optimized higher layer protocol for his system. It is intended for use in stationary or mobile machine systems requiring real-time control, e.g., spinning machines, weaving or knitting machines, saw mills, robots, cranes, excavators, etc. CAN Kingdom supports a machine development philosophy characterized by understandability, safety, simplicity and effectiveness. A key benefit in achieving this is that system designers and module designers to a great extent can work independently of each other. This philosophy opens the way for system designers to find suitable third party modules that can easily be integrated into their systems. It also enforces a conform documentation of modules and systems.

The language used in this document may seem unfamiliar, especially to people with experience in traditional networking. You will, e.g., not find any resemblance with the ISO Open Systems Interconnections (OSI) reference model. The main reason for this is that the OSI model does not support real-time networks! A basic idea in the OSI seven layer model is that the designer should only need to care about two levels; the level where he is working and the interface to the level below. When designing a real-time system however, you always have to care about how long time it takes for transmitting a piece of information *through all layers* from one node to another. How this transmission is made is already mainly given by the CAN protocol in a way that does not fit very well into the OSI layers. CAN deals mainly with the lower OSI layers but the arbitration scheme results in messages priorities, something that belongs to the application layer (ref. [3])! The "Client/Server" model used in OSI also tends leading thoughts into a point to point flow of information, which is contradictory to the message oriented idea of the CAN protocol with broadcasting and multicasting possibilities. These possibilities are a great strength of CAN as they reduce the amount of communication needed in a system. Further more all nodes will get the same information simultaneously with the CAN protocol and this feature is valuable in Controller Area Networks where the nodes have to work in a synchronized way. You do not find anything about this in the OSI model. Thus it does not provide any real help to make an efficient use of the CAN protocol or to make a CAN network understandable. That is why the OSI model is not used here. CAN Kingdom uses a simile of a nation and its postal administration instead. This makes it easy for anyone involved in a Controller Area Network development, whatever his background, to understand the principles of the information flow and the work of CAN Controllers in the system. With this "postal administration" model everybody can clearly see that the problem is not just to transfer information but also the timing of transfers. It takes time to write a letter, to have it mailed and it takes time for the receiver to collect it and to read it. Some additional time is needed to accumulate something to write about and when a letter is read it takes some time to act upon it. This highlights the need for proper timing of message transmissions!

In any Controller Area Network the *timing* of messages is essential. By the bitwise arbitration procedure used in the CAN protocol the bus bandwidth is utilized in the best possible way [ref. 9]. When a collision occurs on the bus, the higher priority message will lose a minimum of time due to the identifier arbitration. Thus, a CAN identifier is not a conventional identifier. It does not only identify a message

but gives it a bus access priority¹. The only one who knows the relative importance between different messages is the **system designer**. Consequently he is the *only one who can assign appropriate identifiers* to the messages! This fact is a cornerstone in CAN Kingdom. Here a **module designer never has to care about CAN identifiers**.

Controller Area Networks are often very complex. In many ways they closely resemble a society. In a well-organized society the production sector as well as the service sector works smoothly and effectively with a small, decentralized, administration. Sub-suppliers deliver correct parts on time to the main suppliers, trains and planes leave on time and the administrative work is minimized. By receiving just short instructions, well-trained people immediately know what to do and they also know how to act when something unexpected happens. A badly organized society is often characterized by a heavy, centralized administration with lots of paperwork and inefficient communication that creates an ineffective production and bad service.

The same is true for a Controller Area Network. If it is not well organized, it will never be efficient! To ensure that the network will have good structure right from the beginning we use the CAN Kingdom simile. The Controller Area Network is described as a kingdom with a capital and cities representing the different modules or nodes in the network. In the kingdom all cities cooperate, each delivering, in proper time, products or services that the kingdom needs. Communication between the cities is made only by letters which are distributed by a postal system. The postal system has some basic rules fitting the CAN protocol. Each city has a post office with a post master who represents a CAN Controller. As there are different CAN Controllers on the market offering different services, the post offices in a CAN Kingdom can also offer different services. The kingdom is governed by the king in the capital and each city is governed by a mayor. To their assistance they have local city administrations that take care of incoming and outgoing mail. In a small city the administration department might be very small offering only a minimum of services but a big city might have a larger administration offering more services, which makes it more versatile.

The CAN Kingdom model makes it easy to see the structure of the system. The system designer defines his kingdom: Each city is given its specific task, receives the required or requested information by mail and sends information in the same way. The correspondence must often be made within tight time limits as many things must be done concurrent with other cities, e.g., blowing nozzles, regulating currents or positioning actuators. Each city can be developed and tested by a separate working team. In tests the rest of the system can be simulated by a computer that, on time, transmits the letters the city should receive and checks that outgoing letters are correct and sent on time.

2.1 Who can use CAN Kingdom?

The CAN Kingdom is aimed for *system builders* who want to use modules from different suppliers and for *module suppliers* wanting to design modules that easily can be adjusted to fit into different systems. Even very simple modules can be designed according to CAN Kingdom at a reasonable cost. To obtain these features CAN Kingdom is highly structured and involves a setup procedure where the modules get

¹ In fact a *CAN identifier* only separates messages on the bus in case of collisions. What part of a message that shall be interpreted as an identifier is always given by a higher layer protocol. A more proper designation of the CAN "identifier field" would be the CAN "arbitration field".

all information they need in order to work properly in the system. If you develop the system as well as all the modules yourself and thus do not need high flexibility, you can still benefit from CAN Kingdom. It will help you to find an efficient network design. In fact, the only difference between your system and a general one is that the services provided in the setup phase can be omitted in your case. The information a module would get during the setup phase is then instead burnt into a memory of the module at production.

Controller Area Networks are most often true real-time, parallel processor systems. Therefore, special care has been taken for real-time and parallel processor problems. Many system builders have chosen CAN as they have found it to be the only "standard" protocol on the market that is well suited for use in machine controls. By the CAN Kingdom the full potential of the CAN protocol can be utilized and some new features will be introduced that enhance the real-time qualities of CAN. These new features do not violate the CAN standard.

2.2 Some characteristics of CAN Kingdom:

1. The system designer decides which CAN identifiers will be used within the system.
2. The system designer can use the full potential of CAN in a true real-time system.
3. The maximum latency time of any message can be predicted.
4. A node designer can design a node that fits into a multitude of systems.
5. The CAN protocol is used according to the ISO 11898 specification but other physical layers are possible. Extended CAN identifiers can be used.
6. When a system is initiated, it runs through a setup phase.
7. During the setup phase a complete higher layers protocol can be built up including data formats from a bit level and upwards, bus management method, a global clock, CAN identifier assignments, etc.
8. In a system there is a kind of master at least during the setup phase.
9. Before the initialization phase begins, each node has to have an identification number, the identifier of the initialization messages and to know the bit-rate of the CAN bus used.
10. During the initialization phase the master assigns proper identifiers to all messages used in the system.
11. The master is able to check which nodes are connected to the network.

12. It is possible to connect any module following the CAN protocol specification (ISO 11898) to a CAN Kingdom network.
13. Not all services provided in CAN Kingdom have to be implemented. Some of them are not always required. The minimum implementation includes that each module will present itself to the network, assign identifiers to messages and respond to a start/stop message from the master.

3. CAN Kingdom symbols.

When developing Controller Area Networks it is essential that everybody involved can easily see what information will be distributed over the network and how this is done. Therefore some special CAN Kingdom symbols and words will be used from now on. The CAN Kingdom words are always Capitalized and when they appear for the first time they will be written in **bold** letters. The paragraph 12 contains a complete CAN Kingdom glossary.

3.1 Hardware symbols.

In the CAN Kingdom simile a complete machine is symbolized with a **Kingdom**. In this Kingdom there is a **Capital** with a **King** responsible for law and order in the Kingdom. The Capital is the hardware of the node that the system designer or **Kingdom Founder** has to develop and the King represents the software of the same node. In the Kingdom there are **Cities**, each governed by a **Mayor**, performing the activities wanted. All information exchanged between Cities in the Kingdom is carried out by the **Postal System**. Each City has a **Post Office** with a **Postmaster**. In short:

* Capital / King	Network Manager
* Postal System	CAN bus and CAN protocol
* City / Mayor	CAN node
* Post Office / Postmaster	CAN Controller

3.1.1 Some remarks about the Postal System and Post Offices.

In a CAN Kingdom the Postal System always gives a peculiar service: Bulk mail out of every Letter sent!

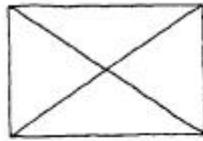
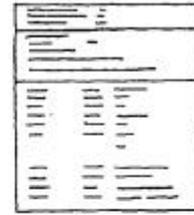
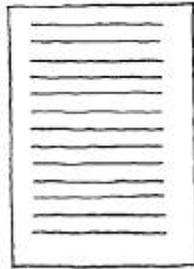
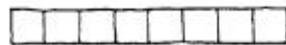
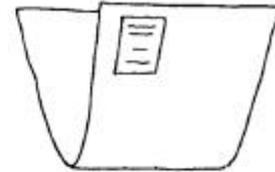
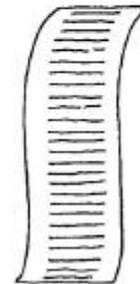
A Post Office corresponds to a CAN Controller and the Postmaster to the services that a particular CAN Controller offers. Some CAN Controllers accept and flag the CPU about all messages received and other CAN Controllers have a register of a limited number of messages that they will receive and flag. In the CAN Kingdom we have corresponding Postmasters that can provide different services. In a small City the Post Office may also be small with a Postmaster that only has one box for incoming and one box for outgoing **Letters** (CAN messages). The Postmaster will receive every Letter sent in the Kingdom (as all Letters are handled as bulk mail) and every time he alerts the Mayor who himself has to check whether the Letter is for him or not. This takes time! The Kingdom Founder has to be aware of this and restrict the amount of Letters sent in the Kingdom so that all Mayors will have time enough to govern their Cities.

Other Post Offices offer more extensive services, e.g., boxes for Letters with certain **Envelopes** (CAN identifiers), time stamping of incoming post, etc. The Mayor will only be interrupted when the Postmaster receives a Letter of importance. Nevertheless, this service may take some extra time at the Post Office and is most often offered only for a limited number of Letters.

In a CAN Kingdom timing is essential. Some Post Offices can offer time stamping of received Letters. When a Letter is correctly received, the Postmaster tags it with the arrival time according to a local clock. I am sure that we soon will see Post Offices providing accurate "Standard Kingdom Time." Then the Mayors have the possibility of time stamping information, e.g., the time when a pressure or temperature was recorded. A receiving Mayor can then take measures to compensate for the time lag in the Postal System and in his own organization.

3.2 Information symbols.

To build up an information structure of the CAN Kingdom we use nine basic building blocks:

Envelopes**Forms****Pages****Documents****Lines****Folders****Bits****Letters****Lists**

All Envelopes are enumerated. An Envelope equals a CAN ID.

A Page can have 0 - 8 Lines. Each Line holds one byte (eight Bits) of data.

A Page equals a CAN data block.

A Line equals a CAN data byte.

A Letter consists of one Envelope containing one Page.

A Letter equals a CAN message.

Forms, Documents, Folders and Lists are used for administration of information within and between the Cities and will be described in detail later on (chapter 13). With these basic building blocks we can build data and message structures from bit level up to a fully controlled, true real-time data exchange. We can

start the building procedure at a suitable level in this hierarchy depending on the complexity and demand of a Kingdom with its Capital and Cities. The higher this level is, the simpler the task is.

Before going into details we will now have a short look at the technique used for setting up information exchange between the Cities in the Kingdom.

3.3 Parameters that have to be defined before a CAN Kingdom setup phase takes place.

Before a Mayor can get in touch with his King, he has to have the following information:

- The physical address number of his City in the Kingdom.
- Actual bit rate and sampling point.
- The Envelope Number of the King's Letter.

How this information should be distributed and stored is not specified in CAN Kingdom. One way to do it is to have a small serial memory placed in the bus connector on the system side, a T-CANnector (ref. [4]).

Throughout this paper it is anticipated that the King's Envelope is number zero but, if the Kingdom Founder so prefers, he can let the King tell the Mayors to exchange this Envelope for another one during the setup phase by a King's Letter. This is a new feature in rev. 3 that opens further possibilities for the system designer. He can build a "united Kingdom" as a federation of several Kingdoms. He might like to have another protocol coexisting on the same bus where this other CAN higher layer protocol already defines the CAN identifier zero for another purpose. Then the King's Envelope has to have another number than zero.

3.4 The setup phase in a simple CAN Kingdom.

Remember now that the Kingdom Founder (the network designer) always has **full knowledge** about the Cities. He knows everything that each Mayor will need to make his City tick. When designing the Capital and creating the King, the Kingdom Founder transfers this knowledge to the King. The situation for the Mayors in the Cities is different: When they are connected to the Postal System, they have no knowledge at all about the Kingdom. However, they know how the Postal System works and how to get their orders from the King.

A CAN Kingdom has always a setup phase. In the beginning of this phase the King has all Envelopes at his disposal. Every Mayor has a set of enumerated Folders. There are Folders for Documents to be transmitted and Folders for Documents to be received. The King has a list of the Folders used in each City. The Envelope 0 is the **King's Envelope** used for the Letters that he sends (at least) during the setup phase (**Figure 2**).

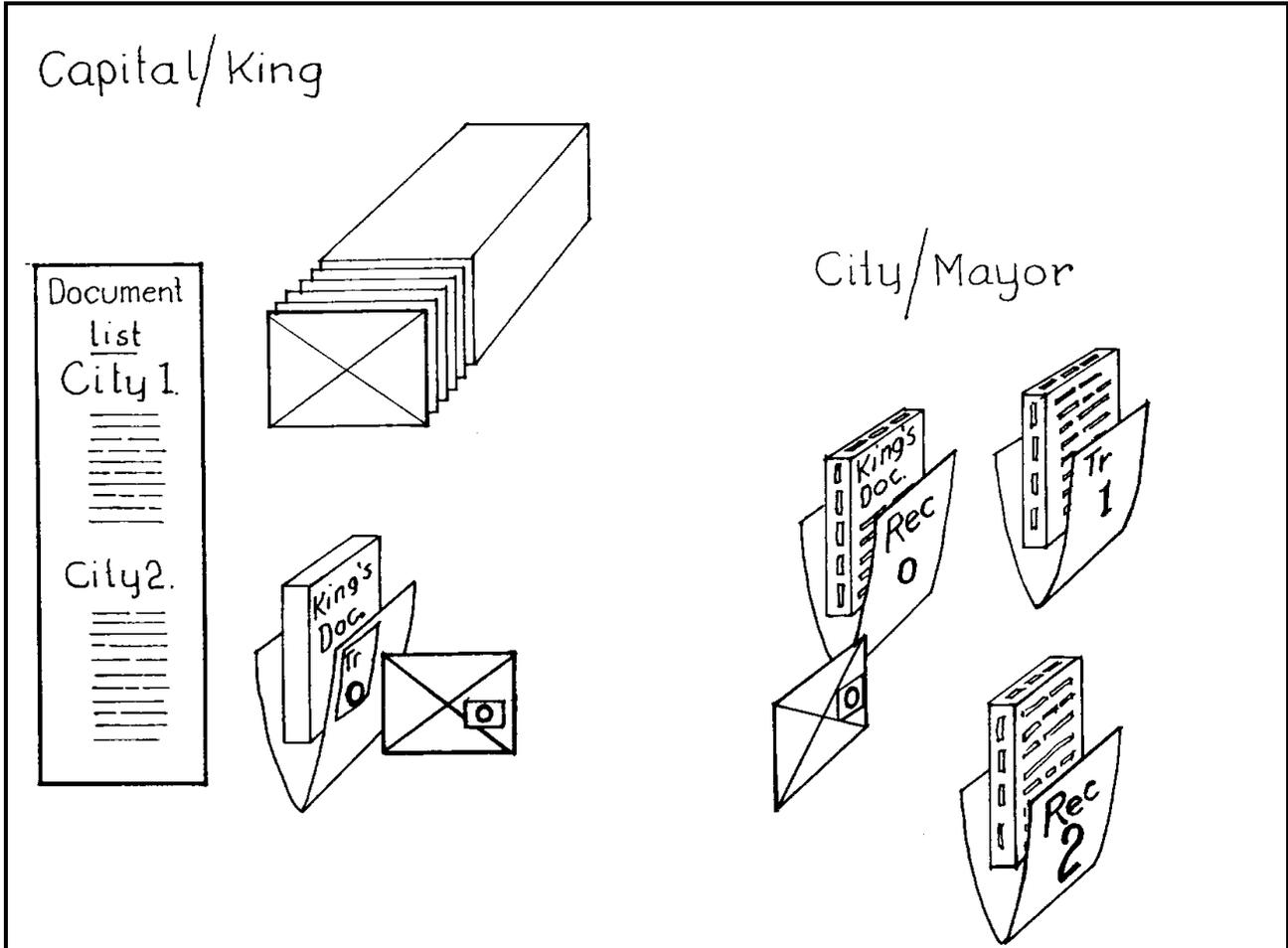


Figure 2. Basic building blocks in the Capital and a City of a simple Kingdom.

By looking in the Document List the King can see what Folder the Mayor of City 1 uses to keep the Form containing Lines (data) to be transmitted. The King can also see which Folder City 2 will use containing a corresponding Form where the same Lines will be placed when received (**Figure 2**).

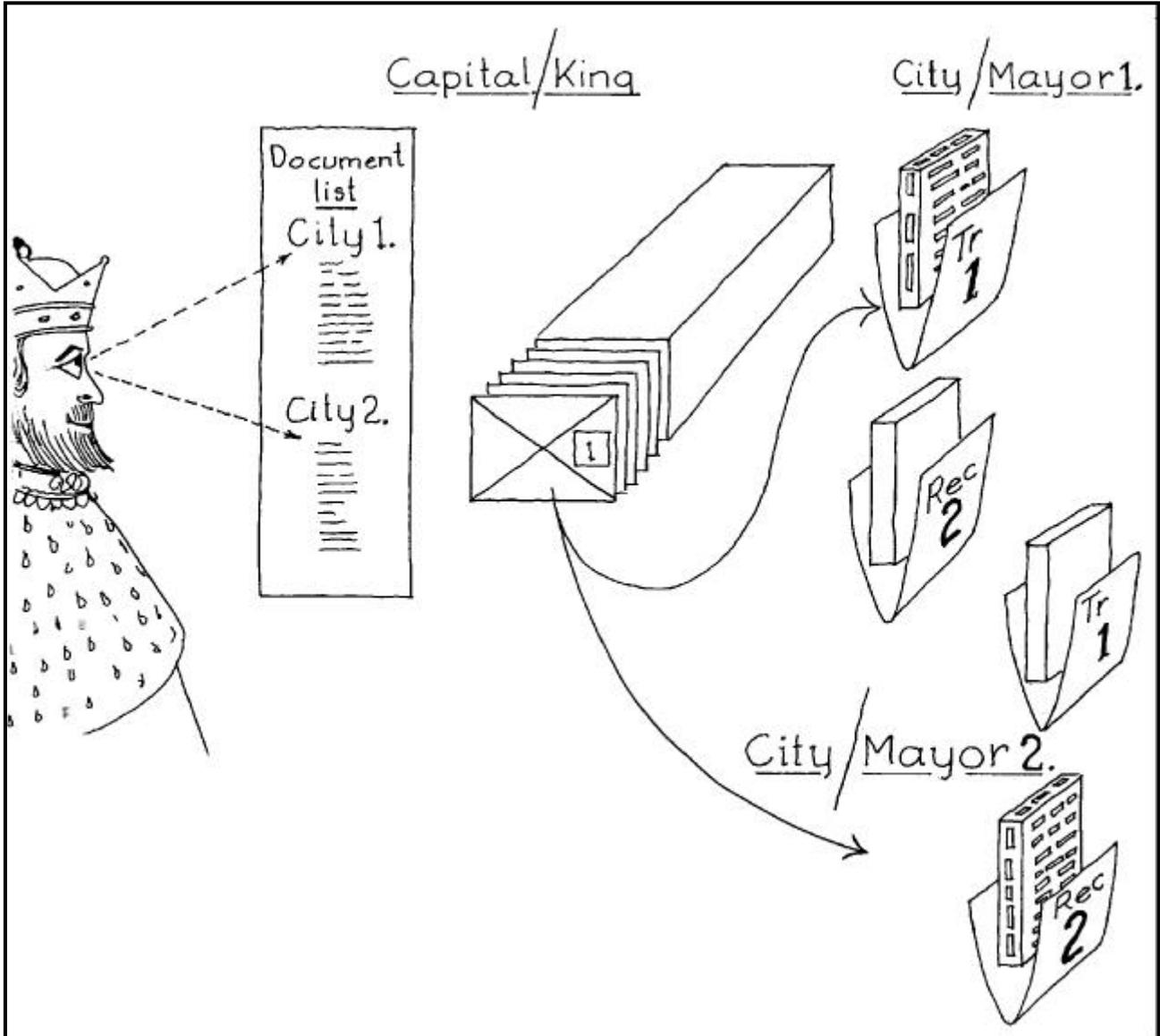


Figure 3. The King assigns Envelopes to Folders enabling two Mayors to communicate.

With a King's Letter the King first assigns a specific Envelope (here Envelope 1) to the Receive Folder at City 2 and then he assigns the very same Envelope to the Transmit Folder of City 1. By this procedure the King can distribute Envelopes to all Cities enabling them to exchange data. He can assign the same Envelope to Receive Folders at several Cities and they will then simultaneously receive the same Letter in one **Round** (CAN message transmission).

3.5 The Letter distribution.

We already know the following:

- * All Envelopes are enumerated. An Envelope equals a CAN ID.

- * A Page can have 0 - 8 Lines. Each Line holds one byte (eight Bits) of data. A Page equals a CAN data block.
- * A Letter, which consists of one Envelope containing one Page, equals a CAN message.

The concept "Letter" is essential in the CAN Kingdom. Letters symbolize CAN messages transmitted over the network. Like all postal services the Postal System of CAN Kingdom does not care about the contents of a Letter as long as it conforms to the rules. A rule in CAN Kingdom is that a Letter always contains only one Page, which can be empty or contain up to eight Lines. At each delivery the **Postman** (a symbol for the CAN bitwise arbitration) asks all the Postmasters if they want to transmit any Letters. Then he takes the Letter that has the lowest Envelope Number and delivers it to *all* Postmasters. This procedure is continuously repeated. The Postman only takes care of just one Letter at each Round and this is the Envelope with the lowest Envelope Number. Thus a Postmaster trying to hand over a Letter with a high Envelope Number may have to try several times before his Letter is accepted for transmission.

The Postmen's simile clearly shows some important characteristics of CAN:

- * The Postman has to visit every Post Office to find out who has the Letter with the lowest Envelope Number. Thus, the longer the **Postal Route** (CAN bus signal path) is, the longer time the Postman needs to complete a delivery.
- * If a Mayor has the Envelope with the lowest Envelope Number for a Letter, he is able, by continuously sending this Letter, to block all other Mayors from having any of their Letters delivered.
- * A Mayor who only has Letters with high Envelope Numbers might never be able to have any Letters transmitted.

Obviously the distribution of Envelopes is very important, especially if several Letters must be delivered within a short time. The Kingdom Founder really has to care about *the timing* of the Letter distribution. Often Mayors have to cooperate in concert and then they must receive their Letters in due time. One way to ensure "just in time" deliveries is to have the Mayors sending Letters in a synchronous way. This will guarantee that the Postal System only has to handle one Letter at a time under normal conditions. Unpredictable events can then be signaled in a safe way by Letters with proper Envelopes that ensures the Letters will be transmitted as soon as no time-critical transmission takes place. Another technique to solve the queuing problem is to restrict the reuse of an Envelope. Different bus management methods are discussed in paragraph 6, page 49.

4. CAN Kingdom building blocks.

4.1 The Form.

CAN Kingdom Letters are very short. The Bits on the Lines are encoded information. When a Mayor wants to send a piece of information to his counterpart in another City, he has to encode it before it can be placed as Lines on a Page. Inversely, when he receives a Letter he has to decode the Lines of the Page to make sense out of it. The tool for this is the "Form." The Form tells where on a Page a certain piece of information should be placed or is expected and in which format it should be presented. A Form explains the meaning of each Line on a Page. Some Lines may carry a measured value. Any measurement is made according to a standard, e.g., a temperature can be presented as degree Centigrade, degree Fahrenheit or Kelvin. The value can be written in different data formats, e.g., as an unsigned byte, 8-4-2-1-BCD, a signed two bytes floating point or as ASCII bytes. A Form for a measurement defines the unit chosen and in what data format the value is presented.

An example of a simple Form for a temperature measuring City is shown below (Document and List information will be explained later in the paragraph 4.2 "The Document" on page 15 and 4.3 "Lists" on page 16):

Form List:	1.1
Form number:	2.1
Document name:	TEMPERATURE
Document type:	Transmit
Page description.	
Number of Lines:	1
Data description:	TEMPERATURE Range: 0 - 255 °C
Line description.	
Line 0.	Temperature in Centigrade. Resolution: 1 °C.
	0 °C = 00000000
Data format:	8-bit unsigned integer

A City Founder can design fixed Forms for his City. As a City may be used in different Kingdoms for different customers or markets, he may have to design several Forms presenting the same information in different ways to make the City attractive. Alternatively he might give the King the possibility to construct Forms that fits his Kingdom out of listed elements. How this is done will be described in the

paragraph 8.1.13 "The **King's Page 17**. Create a Document, a Form or a Line from predefined Form-, Line- or Bit-Lists" on page 77.

If two Mayors in a Kingdom will exchange information, they have to have *exactly matching Forms*. The Forms are thus a vital part of a City specification and a Kingdom Founder has to study them carefully to judge whether a City is a possible choice for his Kingdom or not.

4.2 The Document.

A Letter consists of one Page in one Envelope. This implies that when two Mayors need to exchange more information than can be carried by eight Lines, they have to send more Letters. A convenient way to do this is to use the same Envelope for relating pieces of information. This requires a set of Forms where one Line or at least some Bits of a Line are used for pagination. By reading this Page Number the receiving Mayor can pick up the right Form for decoding the Page. A set of Forms for such Paginated Pages designates a *Document*.

If we use one Line for pagination, we can send $256 \times 7 = 1792$ Lines of information in one Envelope to one or a group of Cities. By using another Line as an address we can send 1536 Lines of individual information to 255 different Cities. This is the technique used in the King's Letter. By using just one Envelope, e.g., the Envelope Number 0, the King can transfer much information to each City. Below is an example of a Form belonging to a Document containing several Forms; the Form of Page 3 of the King's Document in a City:

Form List: 0	
Form number: 3	
Document name: King's Document	
Document type: Receive	
<i>Page description.</i>	
Page number: 3	
Number of Lines: 8	
Data description: The King's Page 3. Assigning a City to Groups.	
<i>Line description.</i>	
Data format of all lines: 8-bit unsigned integer.	
Line 0:	City address
Line 1:	3 (Page Number)
Line 2:	The address of the first Group to which the City is assigned.
Line 3:	Address of the second Group or blank (00000000).
Line 4:	Address of the third Group or blank.
Line 5:	Address of the fourth Group or blank.
Line 6:	Address of the fifth Group or blank.
Line 7:	Address of the sixth Group or blank.

To make the concept "Document" unambiguous it is defined as follows:

*A Document is a set of Forms
describing information carried in one Envelope.*

By this definition a Document may have one or several Forms. The shortest Document possible is one with only one Form containing no Lines, a **Null Document**. This describes a blank Page and a Letter with such a Page can be used for fast trigger signaling. Null Documents in different Cities may be assigned the same Envelopes even for transmission and therefore Null Documents lend themselves for carrying different kinds of alarms.

The concept "Document" is useful for the documentation of a City. The Documents describe what information a City may receive or transmit. In search of suitable Cities for his Kingdom a Kingdom Founder first looks for the Documents of the City specifications. By just looking at the Documents he can see whether a City is appropriate at all or not.

4.3 Lists.

The **City Founder** always knows what information his City must receive and what it can transmit in different situations but frequently he does not know *how* his City will be *used* in a specific Kingdom. ***This is only known by the Kingdom Founder.*** To offer flexibility the City may have several different Documents from which the King can choose. Then the Documents will be numbered and listed in two or more "**Documents List**" in the City documentation.

A City can be even more flexible, offering the King the opportunity to design Documents from predefined Forms. These Forms are then listed in a "**Page Forms List.**"

The same pattern can be used further on: Lines on Forms are then listed in a "**Line Forms List,**" Bits on lines in a "**Bit Forms List.**"

Documents can be of two different types: Receive or Transmit. This is also true for Lines and Bits. To distinguish between these types the Lists are identified with a letter plus a number n:

Receive Lists and their records are identified with R and a number:
R0, R1, R2, . . . , Rn.

Transmit Lists and their records are identified with T and a number:
T0, T1, T2, . . . , Tn.

The King's Letter containing Page 17 (paragraph 8.1.13, page 77) is used for creating new lists and list records from primitives supplied by the City Founder.

Compressed Envelopes (explained in paragraph 4.6) are preferably stored in a "**Compressed Envelopes List.**"

All Lists in a City have an identification number between 0 and 255.

4.4 The Folder.

A Folder is the link between a Document and the Postal System. One or more Envelopes are assigned to a Folder that contains one Document.

The Folders help the City Founders (including the Kingdom Founder) organize their information in a logical way. First two sets of Documents are prepared; one set for information that will be received by the City and another set for information that the City can send. Then these Documents can be put into labeled Folders. A **Folder Label** contains all necessary Postal information for the exchanging of Letters between Cities.

By looking at how the Folders are used we get a clear picture of the flexibility of a City. At the lowest and most rigid level a City has predefined Folders containing predefined Documents. Often this will be quite sufficient. More flexible Cities have Documents that the King can order the Mayor to put into suitable Folders.

Folders can be used to distinguish information that a City *may use or produce* from information that *actually is distributed* within the Kingdom. A City can have a whole library of Documents that makes it possible to use the City in a multitude of Kingdoms. When the City is placed in a specific Kingdom, it will only need a few of the Documents, which are then placed into Folders. The King orders the Mayor what Documents are to be put into which Folders and through this procedure each Document gets an identification, specific to this Kingdom. Matching Documents can be placed into Folders with the same number throughout the Kingdom. The King can then enable or disable the use of a Document by a single broadcast or multicast command to the Mayors, just referring to the Folder number where the Document is placed.

A City can have up to 256 different Folders for incoming or outgoing Documents. The Mayor puts the Documents for the information that he will send or receive in these Folders. The King will then assign Envelopes to Folders containing a Document of interest for the Kingdom. The Folder number 0 and the Folder number 1 are always predefined containing the **King's Document** and the **City Identification Document** respectively.

The advantages of letting the King decide which Documents will be put into which Folders is that matching Documents in different Cities will have a common identification throughout the Kingdom. The disadvantage is that this requires some software and a City Founder may find it cheaper to have the Documents placed in fixed Folders to save memory.

A Folder always has a label, the Folder Label. It contains the following information:

1. Folder Number.
2. Document List Number.
3. Document Number.
4. Transmit / Receive.
5. The CAN Control Field according to the CAN specification.
6. Remote Envelope(s).

An Envelope can be set as "remote" according to the CAN specification by the RTR bit. How RTR set to 1 is interpreted is depending on the application corresponding to the Document in the Folder and has to be defined in the City documentation.

7. Enable / disable the Folder.
By disabling the Folder any CAN communication by the application corresponding the Document in this Folder is interrupted.
8. Envelope(s) assigned to this Folder.
9. Envelope(s) enable / disable.
The use of an Envelope can be switched on or off with an enable/disable tag.

The positions 1 - 7 above are given by the City Founder in the default settings of a Document and/or by the King using the King's Letter, Page 16. The positions 8 - 9 are set by the King's Letter, Page 2. (The King's Letters are defined in paragraph 8.1, "The King's Document.") How much of this information that can be given by King's Letters is depending on the Mayors ability to provide the different facilities. By having matching Documents in Folders with the same number in all Cities the King might easily switch an information flow on or off by multicasting King's Page 2 or 16.

The City Founder may set Folder Labels once and for all (but position 8 and 9 above) on predefined Folders containing specific Documents. In more advanced Cities the Folders with Folder Numbers from 2 to 255 are empty from the beginning. The Mayor has a library of numbered Documents and he will, at the King's command, put a specific Document into a specific Folder.

Some important remarks:

- * A Folder can be empty or contain one and **only** one Document.
- * One or more Envelopes can be allocated to one Folder.
- * An individual Envelope can be set as **transmit** on one and **only** one of the enableed Folders within the entire Kingdom, except when the Folder contains a Null Document.

4.5 The Envelope.

In the ISO CAN standard [5] the specification of the Logical Link Control (LLC) data frame states that an LLC data frame is composed of three bit fields: Identifier field, Data Length Code (DLC) field and LLC data field. The words "Identifier field" and "Data field" are unfortunately used as they might give the reader the impression that he has to use a specific part of a CAN message for identification and another part for data. This misconception can block the mind for finding efficient solutions to many problems. From a general system point of view the identifier field plus the data field can be seen as a set from 11 up to 93 data bits that can be used at will.

A CAN Message contains from 11 up to 93 data bits.

11 (29) Compulsory

0 - 64 Optional

Arbitration

Data only, not *Information!*

These bits are data only and will be turned into information by a higher layer protocol¹. The CAN "identifier field" is an arbitration field and the "data field" is a non-arbitration field. In CAN Kingdom the Envelope is defined as follows:

An Envelope is the arbitration field of a CAN message.

An example when the CAN "identifier field" is only a part of the message identifier is a King's Letter. In a King's Letter the Envelope tells a Mayor that he has received a King's Letter and the first Line on the Page if it is meant for him or not. The second Line tells which Form the addressed Mayor should use to decode the Page. The Envelope plus the two first Lines can be seen as the identifier. Here the identifier is more than the Envelope but often the Envelope is used as a common identifier within a Kingdom.

¹ In CAN Kingdom the DLC field is excluded as a data carrier. The reason for this is that the DLC is used in different ways by different CAN Controllers.

*Data is turned into information by a higher layer protocol,
e.g., composed by CAN Kingdom*

Identifier			Data Frame
King's Envelope	City address	Form number	Information
<i>Envelope</i>	<i>Page</i>		

King's Letter

In CAN Kingdom the Envelopes are divided into two groups:

1. Std. Envelopes, numbered from 0 through 2031.
2. Ext. Envelopes, numbered from 0 through $2032 * 2^{18} - 1$

Priorities:

Std. Envelope 0 has the highest priority followed by a group of 2^{18} Ext. Envelopes, starting with Ext. Envelope 0 through Ext. Envelope 262143, followed by Std. Envelope 1 and so on according to:

Std. N , Ext. N . . . Ext. $(N * 2^{18} - 1)$, Std. N+1, etc. . .

4.6 Compressed Documents and Letters.

The optional CAN Kingdom concept of "Compressed Documents and Letters" allows City and Kingdom Founders to optimize the use of the 11 to 93 data bits in a CAN message. Readers not yet quite familiar with CAN Kingdom can safely omit this paragraph and continue reading on paragraph 4.8, page 39.

In most CAN systems today the arbitration field is used as an identifier and the data field is used for transportation of bytes. However, earlier it has been pointed out a couple of times that a CAN message can be seen as just a bit-stream interpreted by a higher layer protocol. This opens ways to use the available CAN band-width more efficiently and to take advantage of the inherent capacity of some CAN Controllers to act as co-processors. Compressed Letters is the tool for advanced CAN users to squeeze out the maximum power of CAN. A Compressed Letter is characterized in that the data representation

is not based on eight bit bytes and/or that the CAN ID field is used for carrying data. The base for constructing Compressed Letters is a numbering convention for the bits in a CAN message¹:

1. The "BASE ID" (11 bit IDENTIFIER) bits are enumerated with a negative sign from -29 to -19. This is also the numeration of a Std. ID.
2. The "Extended ID" (18 bit IDENTIFIER) bits are numbered from -18 to -1. A complete Ext. ID bits are numbered -29 to -1.
3. The "Data Field" bits are numbered from 0 to 63.

Compressed Letters can be composed in three ways:

1. Compressed Pages in regular Envelopes.
2. Regular Pages in Compressed Envelopes.
3. Fully Compressed Letters.

4.6.1 Compressed Pages in regular Envelopes.

In this case the only difference between the Letters we have discussed so far and a Compressed Letter is how Pages are organized. A Page holds Lines and most often one or more complete Lines are used for carrying the value of a variable, a parameter, etc. This byte oriented presentation is a simple way to carry data as most micro controllers are designed to handle bytes. But sometimes a measured value does not fit exactly into an eight bit byte or a sixteen bit word. Analog values are often presented by ten or twelve bits and a Page would carry more data if the eight bit pattern is abandoned. E.g., in a five-axis machine twelve bit set point values to all axes might be sent in one Letter, saving bandwidth and simplifying coordination. This can be done by the Compressed Page concept.

A City supporting Compressed Pages has a Presentation List where available constants, parameters, variables, etc. and information about them can be found. Then with references to List and Record numbers the King can instruct the Mayor by King's Page 18 (paragraph 8.1.14, page 79) where to place or read specific data on a Page, i.e., creating a **Compressed Page Form**.

Example:

In a Kingdom one City, with the address 5, produces ten bit set-point values to six Servo Cities with the addresses 10 through 15. The City Founder of City 5 offers two ways to transmit the set-point values. As default each value is sent by separate Documents but as an option the Kingdom Founder can create suitable Documents of his own for his Kingdom. To allow the Kingdom Founder to customize Set-Point Documents and Forms the Mayor of City 5 has a Transmit Document List, a Transmit Form List and a Presentation List for transmit values:

¹See also Appendix 1 "A note on Compressed Forms."

Transmit Document List				
List number:	T1			
Document	RTR	Description	Form List / Record number	Fixed Folder number
1.1	Yes	Mayor's Document	T1 / 1.1 - 2.1	1
2.1	No	Set-Point Speed Servo 1	T1 / 3.1	No
3.1	No	Set-Point Speed Servo 2	T1 / 3.1	No
4.1	No	Set-Point Speed Servo 3	T1 / 3.1	No
5.1	No	Set-Point Speed Servo 4	T1 / 3.1	No
6.1	No	Set-Point Speed Servo 5	T1 / 3.1	No

Transmit Form List		
List number:	T1	
Form	Description	Data format
1.1	Mayor's Page 0	
2.1	Mayor's Page 1	
3.1	Individual Servo Set-Point Speed	DUSINT Little Endian
4.1	Servo Set-Point Speed	Compressed Page
5.1	Servo Set-Point Speed	Compressed Page
6.1	Servo Set-Point Speed	Compressed Page
7.1	Servo Set-Point Speed	Compressed Page
8.1	Servo Set-Point Speed	Compressed Page
9.1	Servo Set-Point Speed	Compressed Page

Transmit Presentation List			
List number:		2.1	
Variable	Description	Data format	Size
1.1	Set-Point Speed, servo 1.	USINT	10 bit
2.1	Set-Point Speed, servo 2.	USINT	10 bit
3.1	Set-Point Speed, servo 3.	USINT	10 bit
4.1	Set-Point Speed, servo 4.	USINT	10 bit
5.1	Set-Point Speed, servo 5.	USINT	10 bit
6.1	Set-Point Speed, servo 6.	USINT	10 bit

Here the Kingdom Founder would like all Servo Cities to be updated by the very same Letter to keep all six Servo Cities synchronized. The procedure to do this is to let the King order the Mayor of City 5, by a sequence of the King's Page 18, to create a Form where the set-point of the first servo is placed in the space from bit 0 - 9, the second from 10 - 19 and so on until the sixth servo from bit 50 - 59. Such King's Pages would look as follows (the first two ones):

Document name: King's Document			
Document List:	T1		
Document Number:	0		
Document type:	Transmit		
<i>Page description.</i>			
Page number:	18		
Number of Lines:	8		
Data description:	The King's Page 18. Create a Compressed Form from predefined Lists.		
<i>Line description.</i>			
Line 0:	00000101	City address 5	
Line 1:	00010010	(Page 18)	
Line 2:	00000001	L	List Extension: 1 Transmit
		CCC	Form type: 000 Compressed Page Form
Line 3:	00000010	Source List number 2 (Presentation List)	
Line 4:	00000001	Source record number 1 (Servo 1)	
Line 5:	00000000	Target Form List number 0 (Transmit Form List)	
Line 6:	00000100	Target record number 4 (Compressed Form)	
Line 7:	00000000	Start Position on target: 0 (Bit 0)	

Document name: King's Document			
Document List:	T1		
Document Number:	0		
Document type:	Transmit		
<i>Page description.</i>			
Page number:	18		
Number of Lines:	8		
Data description:	The King's Page 18. Create a Compressed Form from predefined Lists.		
<i>Line description.</i>			
Line 0:	00000101	City address 5	
Line 1:	00010010	(Page 18)	
Line 2:	00000001	L	List Extension: 1 Transmit
		CCC	Form type: 000 Compressed Page Form
Line 3:	00000010	Source List number 2 (Presentation List)	
Line 4:	00000010	Source record number 2 (Servo 2)	
Line 5:	00000000	Target Form List number 0 (Transmit Form List)	
Line 6:	00000100	Target record number 4 (Compressed Form)	
Line 7:	00001010	Start Position on target: 10 (Bit 10)	

The Servo Cities have then to be adjusted accordingly. The Servo Cities have each a set of the same Documents, Forms and Lists:

Receive Document List				
List number:	0			
Document	RTR	Description	Form List / Record number	Fixed Folder number
0	No	King's Document	0 / 0 - 6	0
1	No	Set-Point Speed	0 / 7	No
2	No	Compressed Page Speed Set-Point	0 / 8	No

Receive Form List	
List number:	0
Form	Description
0	King's Page 0
1	King's Page 1
2	King's Page 2
3	King's Page 3
4	King's Page 4
5	King's Page 17
6	King's Page 18
7	Servo Set-Point Speed, DUSINT
8	Servo Set-Point Speed, Compressed Page

Receive Presentation List			
List number:	1		
Variable	Description	Data format	Size
0	Set-Point Speed	USINT	10 bit

The King will send a King's Page 18 to each Servo City so they can make a Compressed Page Form. The Letters to the Cities 10 and 11 would then read as follows and the rest would read accordingly.

Document name:	King's Document		
Document List:	T1		
Document Number:	0		
Document type:	Transmit		
<i>Page description.</i>			
Page number:	18		
Number of Lines:	8		
Data description:	The King's Page 18. Create a Compressed Form from predefined Lists.		
<i>Line description.</i>			
Line 0:	00001010	City address 10	
Line 1:	00010010	(Page 18)	
Line 2:	00000000	L	List Extension: 0 Receive
		CCC	Form type: 000 Compressed Page Form
Line 3:	00000001	Source List number 1 (Presentation List)	
Line 4:	00000000	Source record number 0 (Servo Speed Set-Point)	
Line 5:	00000000	Target Form List number 0 (Receive Form List)	
Line 6:	00001000	Target record number 8 (Compressed Form)	
Line 7:	00000000	Start Position (LSB) on target: 0 (Bit 0)	

Document name:	King's Document		
Document List:	T1		
Document Number:	0		
Document type:	Transmit		
<i>Page description.</i>			
Page number:	18		
Number of Lines:	8		
Data description:	The King's Page 18. Create a Compressed Form from predefined Lists.		
<i>Line description.</i>			
Line 0:	00001011	City address 11	
Line 1:	00010010	(Page 18)	
Line 2:	00000000	L	List Extension: 0 Receive
		CCC	Form type: 000 Compressed Page Form
Line 3:	00000001	Source List number 1 (Presentation List)	
Line 4:	00000000	Source record number 0 (Servo Speed Set-Point)	
Line 5:	00000000	Target Form List number 0 (Receive Form List)	
Line 6:	00001000	Target record number 8 (Compressed Form)	
Line 7:	00001010	Start Position (LSB) on target: 10 (Bit 10)	

4.6.2 Regular Pages in Compressed Envelopes.

The main task of the Postal System is to distribute bit patterns within the Kingdom in a safe way. In a City the Postmaster may also help the Mayor by sorting out specific Letters of interest to him. This is done by picking out Letters with specific Envelope Numbers or within ranges of Numbers. Most Kingdoms need only a limited amount of Letters for their communication and this makes it possible to let the Envelope Number carry more information than just an identification number. For a Postmaster the Envelope Numbers are just numbers for identification but the Mayor might draw conclusions of the numbers according to a higher layer protocol. For example, the numbers may be encoded to carry

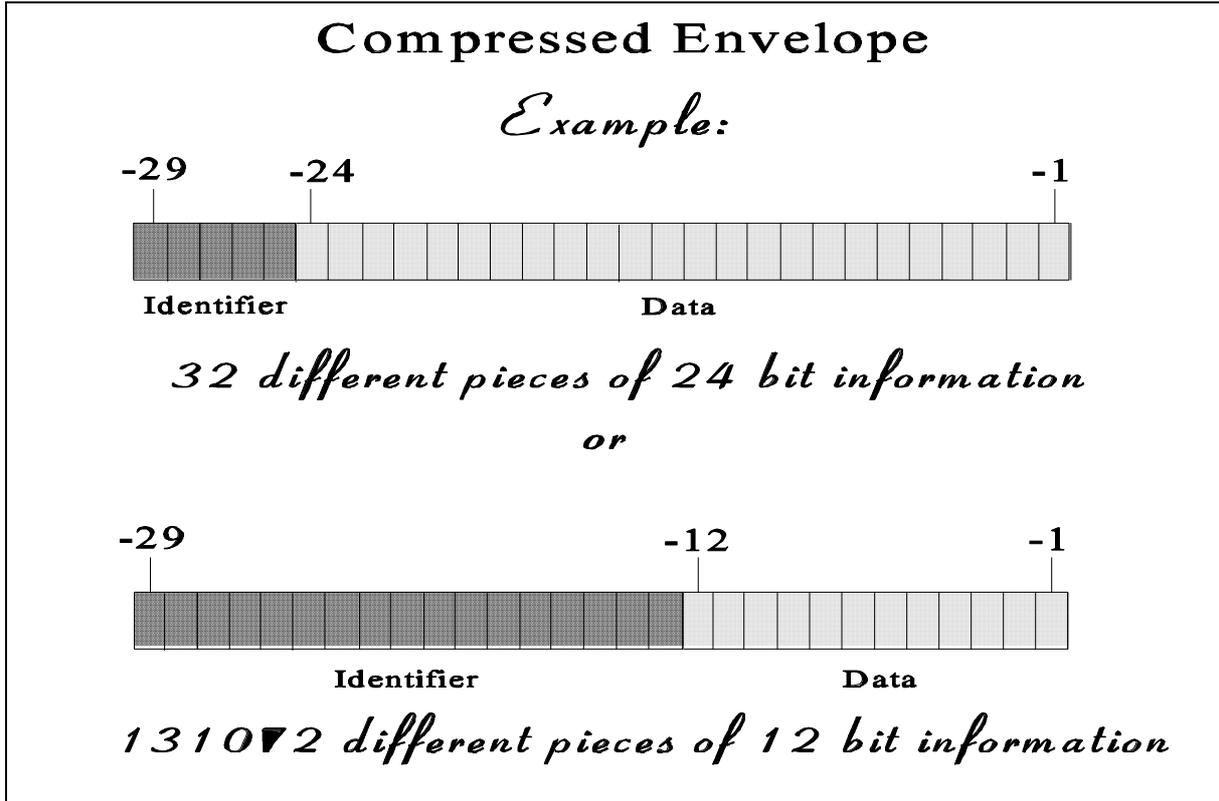
information about the origin of the Letter and about the contents of the Page inside. In fact, some higher layer protocols like SDS and DeviceNet make use of this possibility. However, whatever information the Envelope might carry, the decoding process always starts with picking out the Envelopes associated with a certain Folder.

Sometimes implementing a communication protocol where the Envelopes are constructed out of variables, parameters, addresses, etc. is desirable for a Kingdom Founder. Ext. CAN provides 532,678,640 different identifiers that can be used in miscellaneous ways. Only the Kingdom Founder knows how they can be utilized most efficiently in his Kingdom. To let CAN Kingdom take advantage of the remaining bandwidth within Envelopes, we introduce the concept **Compressed Envelopes** and Forms for such Envelopes. A **Compressed Envelope Form** is a key to decode the negative bits according to the numbering rules of the bits, i.e., the Envelope of a Letter. The first step in decoding an incoming Letter is to associate a Letter to a Document in a Folder. Previously we have used the whole Envelope Number for this purpose but now we will assign only a part of it for this function. To pick out and associate a certain Letter with a Folder a receiving Mayor needs to know a **First Identifier** of that Letter but to fully decode a Letter a Mayor may need additional identifiers. One example is a King's Letter: The First Identifier is the King's Envelope, the second identifier is the address on the first Line and the third identifier is the Page Number on the second line. A receiving Mayor has to have these three ID:s in order to decode a King's Letter. When Compressed Envelopes are *not* used a First Identifier equals an Envelope. Theoretically a First Identifier can be constructed from bits scattered all over an Envelope or even a Letter but in CAN Kingdom the possibility is restricted to only **bits in the arbitration field**.

The origin of a First Identifier can be a fixed number, a parameter value or a variable value from the transmitting City that matches the whole alt. a part of a fixed number or a parameter at the receiving Cities. The Mayor of the *transmitting* City does not have to know what part of an Envelope that holds a First Identifier. In fact, different receiving Mayors may use different parts of a very same Letter as a First Identifier. By King's Letter 19 (page 81) the King constructs a filter that the receiving Mayor will use to identify the Compressed Envelope and assigns it an identification number. This number will be used in King's Letter 2 (LSB of the Envelope Number and the "Compressed flag" set) when assigning the Compressed Envelope to a Folder. The filter is a software filter, used only for Compressed Envelopes. If the Mayor will have a filter for his Post Office, then he will get it by King's Page 20.

Most often an Envelope with a fixed number is assigned to a Folder. Sometimes however, a Mayor can use the services of a Postmaster in a more efficient way if variables are included in the Envelope number. Especially the extended format of the CAN Identifier lends itself to this technique. From the viewpoint of a Mayor the only special thing about an Envelope compared with Lines on a Page is that the Envelope is compulsory in any information exchange and that some Postmasters offer some sorting services. Some data might be written on the Envelope. This can be used for increasing the bandwidth of the communication and/or provide the possibility for a Postmaster to help the Mayor of the City to filter out essential information.¹

¹ Hint: In the CAN standard 2B Std. and Ext. identifiers form two interlaced groups of Envelopes. Within these groups some of the Ext. identifiers can be used for carrying time critical data (which may be written on the Envelope) and the Std. identifiers for all other Letters in a system.

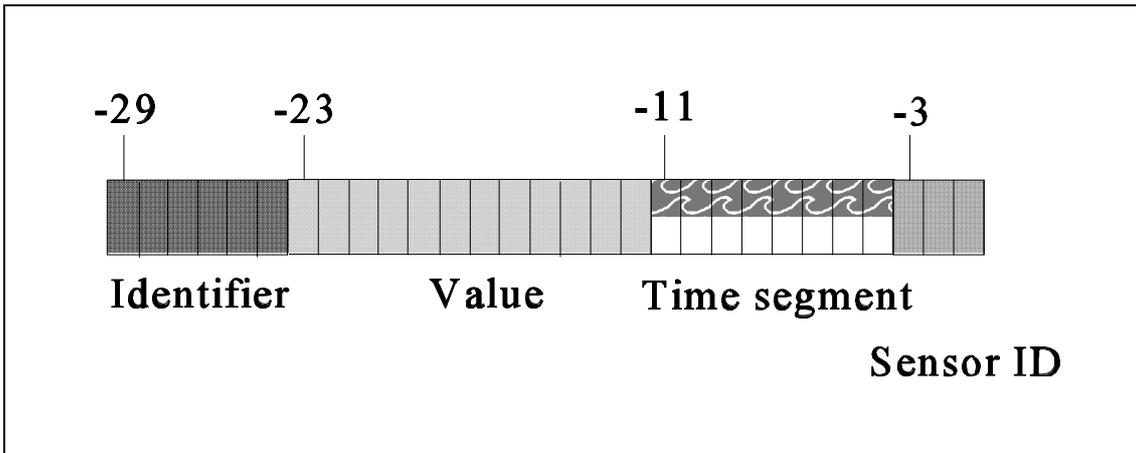


Two examples on how to organize a Compressed Envelope (extended CAN ID):

1. In a specific Kingdom there are less than 63 variables to be measured, all with a 12 bit resolution. Some variables are measured by more than one but at the most eight sensors for redundancy. The measurements are done within one Time Revolution and are time stamped with a time resolution of one Time Segment¹. A Compressed Envelope may look as follows:

Bit number	Content
-29 - -24	Identifier of measured variable.
-23 - -12	Measured value.
-11 - -4	Time Segment
-3 - -1	Sensor identifier.

¹ CAN Kingdom Time specifications are described in chapter 5.



In which order the building blocks will appear on the Envelope is depending on the character of the higher layer protocol, e.g., if it is event driven or time driven, if the bit wise arbitration is used or not, filtering requirements of the Compressed Envelope, etc. The order below might fit better:

Bit number	Content
-29 - -22	Time Segment.
-21 - -16	Identifier of measured variable.
-15 - -13	Sensor identifier.
-12 - -1	Measured value.

2. A Compressed Envelope can be used to let the Postmasters of the Cities filter out trigger signals from a stream of Letters:

In a Kingdom one City, the Sensor City, takes care of measuring the position, speed and acceleration of the main shaft in a machine. (A Sensor City can measure acceleration and speed with a much higher resolution in time and position than any other City can do by derivating position values received by Letters.) Other Cities initiates tasks at certain positions and these positions are depending on the speed of the shaft. When the shaft is rotating, the Sensor sends a Letter at every degree that looks as follows:

Envelope:

Bit	Value
-29 - -21	0 . . . 0
-20 - -19	11
-18 - -11	0 . . . 0
-10	1
-9 - -1	Position (degree), 9-bit unsigned integer.

A Four-Line Page with the following information:

Line 1	Speed	LSB
Line 2	Speed	MSB
Line 3	Acceleration	LSB
Line 4	Acceleration	MSB

A City may be designed so that when the Postmaster recognizes an incoming Letter representing a specific position, the very same signal will directly release a physical activity, e.g., the opening or closing of a valve. The Mayor reads the Page and according to this information he might order the Postmaster to filter out another Envelope at the next time, i.e., another position to trigger the activity. Such procedures are discussed more in detail in ref. [6] and [7]. Bit -20 and -19 being two recessive 1:s leaves four Std. Envelopes and bit number -10 being a recessive 1 allows for a range of up to 256 Ext. Letters with higher priorities that can be used for high priority Letters as alarm messages etc.

4.6.3 Fully Compressed Letters.

The ultimate compression is to use the whole Letter as an entity by starting a data field on the Envelope and let it slip over into the Page. The King's Letter 18 (page 79) offer the possibility for construction and identification of Forms for Fully Compressed Letters.

4.7 Filters.

Filters are needed for two purposes, for extracting First Identifiers and for a Postmaster to filter out Letters of interest to his Mayor. In the first case we have to solve the problem in software but in the second case some CAN Controllers offer hardware assistance. The King's Page 19 is intended for constructing software filters and the King's Page 20 is intended for constructing filters suitable for different Post Offices.

4.7.1 Filters for extracting First Identifiers.

King's Page 19 on page 81 is used defining software message filters in the Cities and such filters are suitable for extracting First Identifiers. Each City implements a filter as a set of *filter primitives*. A primitive represents an arbitrary arithmetic operation on the arbitration field of the incoming CAN messages, such as logical AND, logical OR, invert, compare and so on. Up to eight filter primitives can be used in one filter (however, not every module must support that many). Filter primitives are activated, one at a time, by means of the King's Page 19. The numbering and exact meaning of the filter primitives will vary from City to City and is to be described in the City's documentation.

The purpose of a filter is to extract the First Identifier from an incoming Letter, and then determine if this First Identifier identifies a Folder in the City. By applying the Filter Primitives to an incoming Letter, the Mayor will be able either to reject or to accept the Letter.

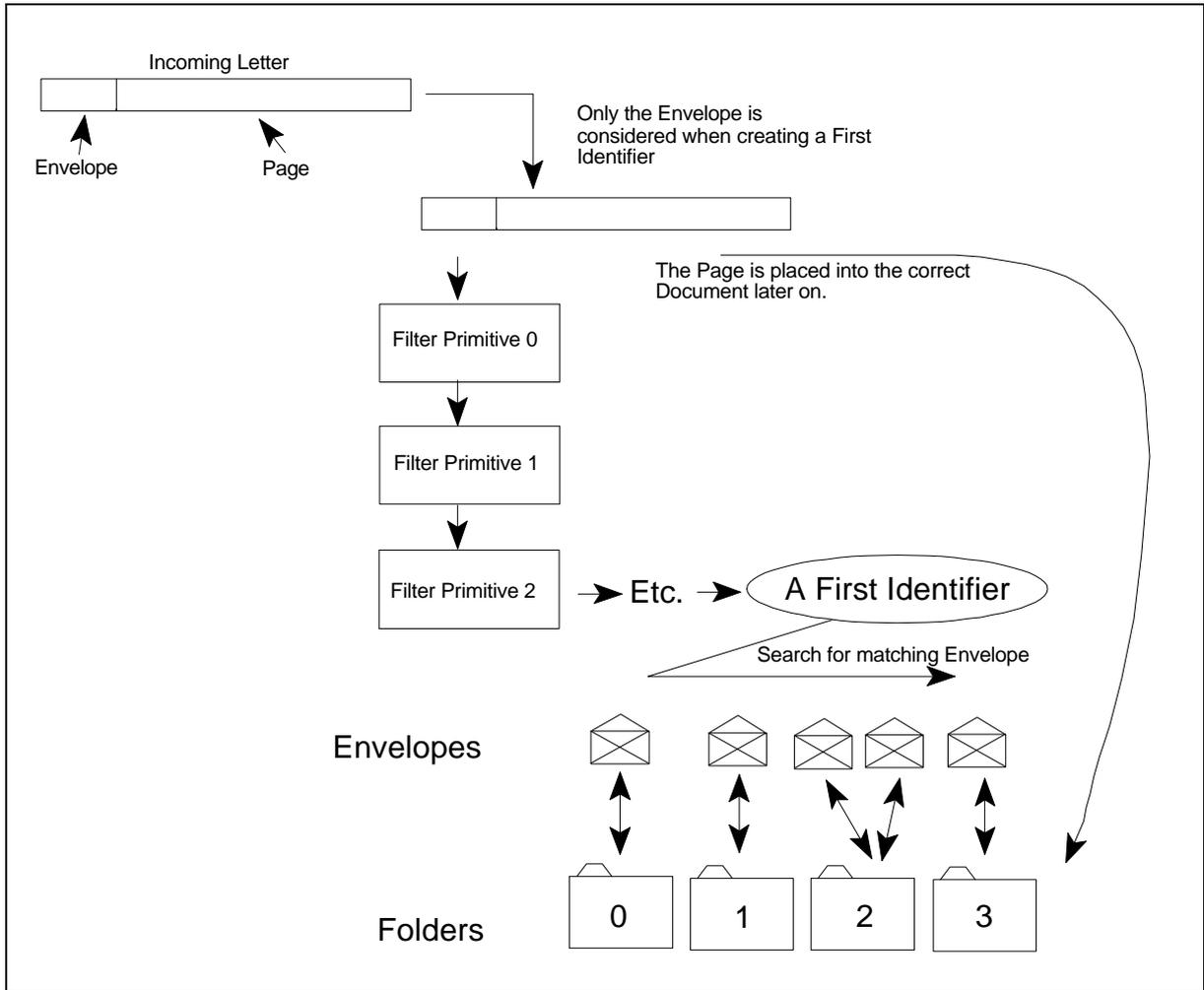


Figure 5: Constructing a First Identifier when a Letter arrives.

Compressed envelopes are not created for reception only. The City Founder should therefore ensure that the operations defined by the filter primitives are reversible, so that they also can be used to *construct* a valid CAN arbitration field, not merely decompose one.

Two examples of basic filter primitives are:

- An AND mask: the “bit pattern” field of King’s Page 19 contains ones in the positions which should be subject to further operations.
- A COMPARE EQUAL value; if the value from a previous operation is equal to the bit pattern, the result of this primitive is “true,” else “false.”

When a Letter arrives at the City, the Mayor can behave as follows:

- For each Folder, apply its filter as specified by previously sent King’s Pages 19 and 2. If the Folder does not have any Compressed Envelopes assigned to it, behave as if the filter were just a COMPARE value of the standard envelope(s) assigned to this folder.
- When applying the filter, first calculate the value of (CAN arbitration field) AND (AND mask value), and then the boolean value of (value from first operation) EQUALS

(COMPARE's "bit pattern"). If the result is "true," then the Letter belongs to the Folder in question; if not, continue the search.

When requested (e.g., by an Action/Reaction Page) to send the contents of a Folder to which a Compressed Envelope is assigned, the Mayor must compose a valid CAN arbitration field from both the data to be written on the Envelope and the data contained in the Filter, as the latter restricts the possible CAN id values. When only the above two filter primitives are used, this is done simply by taking the data, then perform a logical AND with the *inverse* of the AND mask followed by a logical OR with the COMPARE bit pattern. A more complicated filter may give the City Founder a considerable headache, not to mention the increased burden on the Mayor.

A simple example:

Assume the Kingdom Founder wants a City to receive a certain 8-bit value in the arbitration field of a standard CAN message. The remaining 3 bits will be used as identifier field and must equal 100_b . The King will then have to send the following King's Pages to the City in order to initialize it:

1) King's Page 18:

- Line 2: L=1, CCC=001 (List Extension: Transmit; Compressed Envelope, Standard)
- Line 3: Source List = 1
- Line 4: Source record number = 0
- Line 5: Target Form List = 1 (arbitrary)
- Line 6: Target record number = 0 (arbitrary)
- Line 7: Start position = -29

This places variable "Position" in the CAN Arbitration Field at position -29..-22 (the first 8 bits).

2) King's Page 17:

- Line 2: 000 10 00 1 (Form List; Transmit)
- Line 3: Source List number = 1 (same as above)
- Line 4: Source record number = 0 (same as above)
- Line 5: Target List number = 1 (arbitrary)
- Line 6: Target record number = 0 (arbitrary)
- Line 7: Position on Target = 0 (as the document is not paginated)

This constructs a document containing the previously constructed form.

3) King's Page 16:

- Line 2: Folder number = 2 (arbitrary)
- Line 3: 10000000 (no RTR, DLC = 0)
- Line 4: 11010001 (Enable folder, Insert document, Transmit)
- Line 5: Document List number = 1 (same as above)
- Line 6: Document record number = 0 (same as above)
- Line 7: 0 (reserved)

This places the document constructed above in Folder number 2.

4) King's Page 19:

- Line 2-5 : bit pattern = 00000000 00000000 00000111 11111000
- Line 6: Filter Primitive number = 0
- Line 7: Filter number = 0 (arbitrary)

This starts the building of filter number 0. According to its documentation, this City defines primitive 0 as an AND mask.

5) King's Page 19:

Line 2-5 : bit pattern = 00000000 00000000 00000000 00000100

Line 6: Filter Primitive number = 1

Line 7: Filter number = 0 (same as above)

This continues the building of filter number 0. This City defines the filter primitive 1 as a COMPARE EQUAL value.

6) King's Page 2:

Line 2-5: Envelope number = 01000000 00000000 00000000 00000000

Line 6: Folder nr = 2 (same as above)

Line 7: 00000011 (Assign and Enable)

This assigns the envelopes matching the just built filter 0 to folder number 2.

When the setup phase is terminated, the City will be able to filter out all messages with the 8 most significant bits set to any value and the lowest 3 bits set to 100_b.

A filter for extracting the First Identifier of example 2 of Compressed Envelopes on page 31:

Receive Presentation List			
List number:	1		
Variable	Description	Data format	Size
0	Speed	USINT	16 bits
1	Acceleration	USINT	16 bits
2	Position	USINT	8 bits

Filter Primitive List	
Filter Primitive	Description
0	The bit pattern is used for a logical AND operation
1	The bit pattern is used for a COMPARE EQUAL operation

Assuming the City in question has the Presentation List and a Filter Primitive List as above, the King sends the following King's Pages to the module:

1) King's Page 18:

- Line 2: L=1, CCC=110 (List Extension: Transmit; Compressed Letter, Extended)
- Line 3: Source List = 1
- Line 4: Source record number = 0
- Line 5: Target Form List = 1 (arbitrary)
- Line 6: Target record number = 0 (arbitrary)
- Line 7: Start position = 0

This places the variable "Speed" on Form 1, Form List 1, at position 0 (which is the very beginning of the data field).

2) King's Page 18:

- Line 2: L=1, CCC=110 (List Extension: Transmit; Compressed Letter, Extended)
- Line 3: Source List = 1
- Line 4: Source record number = 1
- Line 5: Target Form List = 1 (same as above)
- Line 6: Target record number = 0 (same as above)
- Line 7: Start position = 16

This places the variable "Acceleration" on the same Form at the beginning of Line 2.

3) King's Page 18:

- Line 2: L=1, CCC=110 (List Extension: Transmit; Compressed Letter, Extended)
- Line 3: Source List = 1
- Line 4: Source record number = 2
- Line 5: Target Form List = 1 (same as above)
- Line 6: Target record number = 0 (same as above)
- Line 7: Start position = -9

This places variable "Position" in the CAN Arbitration Field at position -9..-1 (the last 9 bits).

4) King's Page 19:

- Line 2-5 : bit pattern = 10011111 11111111 11111110 00000000
- Line 6: Filter Primitive number = 0
- Line 7: Filter number = 0 (arbitrary)

This starts the building of filter number 0. According to its documentation, this City defines primitive 0 as an AND mask.

5) King's Page 19:

- Line 2-5 : bit pattern = 10000000 00001100 00000010 00000000
- Line 6: Filter Primitive number = 1
- Line 7: Filter number = 0 (same as above)

This continues the building of filter number 0. This City defines the filter primitive 1 as a COMPARE EQUAL value.

6) King's Page 17:

- Line 2: 000 10 00 1 (Form List; Transmit)

- Line 3: Source List number = 1 (same as used above)
- Line 4: Source record number = 0 (same as used above)
- Line 5: Target List number = 1 (arbitrary)
- Line 6: Target record number = 0 (arbitrary)
- Line 7: Position on Target = 0 (as the document is not paginated)

This constructs a document containing the previously constructed form.

7) King's Page 16:

- Line 2: Folder number = 2 (arbitrary)
- Line 3: 10000100 (no RTR, DLC = 4)
- Line 4: 11010001 (Enable folder, Insert document, Transmit)
- Line 5: Document List number = 1 (same as above)
- Line 6: Document record number = 0 (same as above)
- Line 7: 0 (reserved)

This places the document constructed above in Folder number 2.

8) King's Page 2:

- Line 2-5: Envelope number = 01000000 00000000 00000000 00000000
- Line 6: Folder nr = 2
- Line 7: 00000011 (Assign and Enable)

This assigns the envelopes matching the just built filter 0 to folder number 2.

9) Possibly more King's Pages, and at last, King's Page 0: run!

The Sensor City will not continuously measure the position of the rotating shaft. In order to send the Letters, the Mayor must do as follows:

- Take the shaft position value and perform logical AND with the inverse of the bit pattern specified in filter 0, primitive 0.
- Perform logical OR with the bit pattern in filter 0, primitive 1.
- Construct and send a CAN message with the calculated value in the arbitration field and the measured speed and acceleration in the data field.

The City receiving these messages (which has been setup by the King in a similar way) will then interpret the incoming Letters according to the constructed Filter. All Envelopes with an arbitration field of 00000000011000000001xxxxxxxx will match the mask and taken care of according to the Document placed in Folder 1.

4.7.2 Hardware filters.

The principles behind King's Page 20 on page 82 are the same as for King's Page 19: a filter is composed of one or more filter primitives, each representing some kind of arithmetic operation on the arbitration field in an incoming CAN message. There are some differences, however:

- The filter is implemented in hardware.
- The exact meaning of the filter primitives is defined by the hardware used.
- The filter is used only when receiving messages.
- The purpose of the filter is to decrease the Mayor's workload.

The reason for the King - as opposed to the Mayor - providing the mask values is that the Mayor does not know in advance which Envelopes to use. At run time, the Mayor is usually too busy to calculate appropriate mask values. By King's Page 20, the King can set suitable mask values in the Cities. Of course, an advanced City may calculate and set its own masks.

The numbering of the filters and filter primitives is still in the hands of the City Founder. Below are recommendations for some different CAN controllers (please see the respective data sheet for further details):

Intel 82526:

Intel 82527:

Filter 6 (any primitive): The bit pattern specified on Line 2 will be placed in register 06H. The bit pattern specified on Line 3 will be placed in register 07H.

Filter 8 (any primitive): The bit patterns on Line 2 - 5 will be placed in register 08H - 0BH, respectively.

Filter 12 (any primitive): The bit patterns on Line 2 - 5 will be placed in register 0CH - 0FH, respectively.

Philips 82C200 :

Any filter, any primitive: The bit pattern on Line 2 will be placed in the Acceptance Code Register. The bit pattern on Line 3 will be placed in the Acceptance Mask Register.

NEC mPD72005:

Filter 0: The bit patterns on Line 2-5 will be placed in the MKA00-MKA03 registers, respectively.

Filter 1: The bit patterns on Line 2-5 will be placed in the MKA10-MKA13 registers, respectively.

Filter 2: The bit patterns on Line 2-3 (std CAN) or 2-5 (ext. CAN) will be placed in the identifier part of the buffer designated as global buffer 0 (by register GM0).

Filter 3: The bit patterns on Line 2-3 (std CAN) or 2-5 (ext. CAN) will be placed in the identifier part of the buffer designated as global buffer 1 (by register GM1).

4.8 The use of Documents, Folders and Lists. A simple example.

We have a small Kingdom with a Capital and three Cities (**Figure 6**). Two of the Cities do only measure a temperature and the third one needs for its task the temperature measured of the other two.

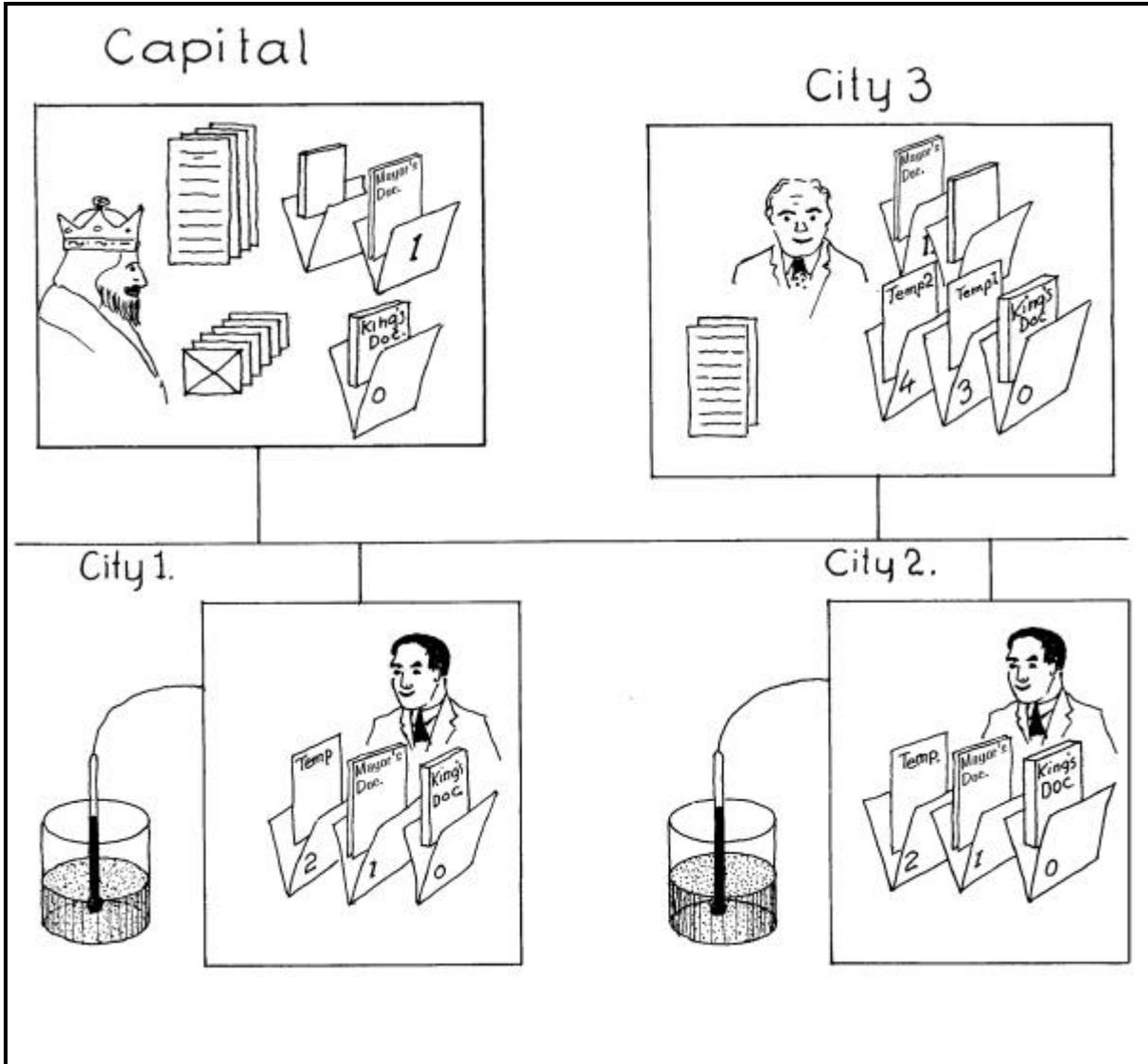


Figure 6. A small Kingdom.

The temperature measuring Cities are very simple. Such a City has only three documents: The King's Document, The City Identification Document and the Temperature Document. (The King's Document and the City Identification Document will be discussed later.) These Documents were put into Folders by the City Founder when he designed the City, so the *Mayor* of the City does not need any Lists. The Lists should any how be included in the City documentation as they will be the first thing a Kingdom Founder would look into when searching for suitable Cities. By looking into the Lists a Kingdom Founder immediately gets an first impression of the City, finding out whether it is an alternative for his Kingdom or not. The formal documentation of the temperature measuring City is shown below:

Document Lists:

Receive Document List				
List number: R0				
Document	RTR	Description	Form List . Record number	Fixed Folder number
0	No	King's Document	R 0.0 - 2	0

Transmit Document List				
List number: T0				
Document	RTR	Description	Form List . Record number	Fixed Folder number
0	Yes	Mayor's Document	T0.0 - 1	1
1	Yes	Temperature, 0 - 255 °C	T0.2	2

Form Lists:

Receive Form List	
List number: R0	
Form	Description
0	King's Page 0
1	King's Page 1
2	King's Page 2

Transmit Form List	
List number: T0	
Form	Description
0	Mayor's Document Page 0
1	Mayor's Document Page 1
2	Temperature Page

Form T0.2:

Form List:	T0
Form number:	2
Document name:	TEMPERATURE
Document Number:	T0.1
Document type:	Transmit
Page description.	
Number of Lines:	1
Data description:	TEMPERATURE Range: 0 - 255 °C
Line description.	
Line 0.	Temperature in Centigrade. Resolution: 1 °C. 0 °C = 00000000
Data format:	8-bit unsigned Integer.

When designing this type of City the City Founder decided that the City is to measure a temperature in the range 0 - 255 °C and transmit the measured temperature as an unsigned integer byte. The Transmit Document List tells the Kingdom Founder that this City will mail a Letter with a measured temperature on request (RTR, remote transmit Yes) and the Form of the Temperature Document that the Mayor will write the measured temperature on Line 0 of the Page as an 8-bit unsigned integer. As the City only produces measurement of one temperature, it is enough with one Document for the production result.

As can be seen in the Lists the City also has a Document R0.0 containing selected Forms of the King's Document and a Document T0.0 containing the Mayor's Document Form, not shown here.

The Mayor of the third City will receive Letters from both the other two Cities and must distinguish between temperature measured at each of them. Then he needs (among other Documents) two Temperature Documents. The Receive Document List and the Temperature Forms of the City 3 may look as below:

Receive Document List				
List number: R0				
Doc.	RTR	Description	Form List Record no.	Fixed Folder number
0	No	King's Document	R0.0 - 6	0
1	No	Temperature Setpoints	R0.7 - 8	No
2	No	Temperature 1	R0.9	No
3	No	Temperature 2	R0.9	No

Temperature Form	
Form List:	R0
Form number:	9
Document name:	TEMPERATURE 1 TEMPERATURE 2
Doc. Number:	R0.2, R0.3
Document type:	Receive
Page description.	
Number of Lines:	1
Data description:	TEMPERATURE Range: 0 - 255 °C
Line description.	
Line 1.	Temperature in Centigrade. Resolution: 1 °C. 0 °C = 00000000
Data type:	8-bit unsigned integer

When the Kingdom Founder, during the design phase, compared the Temperature Document Forms of these Cities, he found that they match each other perfectly. Thus it was clear that it would be possible for the Mayor at City 3 receiving the temperatures from the Mayors at City 1 and 2. The temperature is presented on the same Line and with the same data format in each Form. The Kingdom founder can now make the instructions for his King: The King will order the Mayor of City 3 to place Document R0.2 into Folder 3 and 4 respectively (by King's Page 16). Then the King will (by King's Page 2) assign the same Envelope (e.g., Envelope 15) to the Folder 3 now containing Document R0.2 (TEMPERATURE 1) at City 3 and to the Folder containing Document T0.1 (TEMPERATURE) at City 2. The City 3 can now receive TEMPERATURE 1. By the same procedure the King will link the Document R0.3 (TEMPERATURE 2) at City 3 to Document T0.2 at City 2 by another Envelope, e.g.,

Envelope 16. During the runtime phase the Mayor 1 sends his **TEMPERATURE** in Envelope 15 and Mayor 2 sends his **TEMPERATURE** in Envelope 16. The Mayor of City 3 receives his variables **TEMPERATURE 1** in Envelope 15 and **TEMPERATURE 2** in Envelope 16.

By this example we can clearly see that the Mayors only have to care about their own Cities. They do not have to think about the Kingdom other than obeying the orders of the King. During the foundation of the Kingdom the Kingdom Founder instructs the King to give proper orders for an efficient organization of the country. The Mayors in the different Cities will then work together in a productive way. This is a great strength of the CAN Kingdom approach.

4.9 Summary.

- * The Kingdom Founder creates a CAN higher layer protocol suitable for his Kingdom.
- * The Postman only carries bulk mail on the Postal Route.
- * The Postal System secures that a mailed Letter is correctly received.
- * At each Round on the Postal Route the Postman takes the Letter with the lowest Envelope number.
- * Post Offices may offer different services, e.g., sorting of incoming bulk mail, time stamping of Letters etc.
- * In a CAN Kingdom the King distributes all Envelopes to the Mayors, except the King's Envelope and, if the Base Number is known a priori by the Mayors, the Mayors' Envelopes.
- * Standard Cities can be used in a CAN Kingdom but it is in the hands of the Kingdom Founder to authorize which to be used in his Kingdom.
- * CAN Kingdom facilitates the *documentation* of a City or Capital by specified building blocks, one group describing CAN messages and another group with administration tools. (See boxes below.)
- * A Document has one or more Forms and each Form will have a heading notifying which Documents it belongs to. If a Document has more than one Form, each Form has to be Paginated.

CAN message description:

- 1 **Envelopes.**
 CAN ID:S
- 2 **Pages.**
 Data Fields
- 3 **Lines.**
 Data Bytes
- 4 **Bits.**
 Data Bits
- 5 **Letters.**
 CAN Data Frames or
 CAN Remote Frames

Administration tools:

- 6 **Forms.**
 Page descriptions
- 7 **Documents.**
 Sets of Forms
- 8 **Folders with
 Folder Labels.**
 Linking Documents
 to Envelopes
- 9 **Lists.**
 Organizing information:
 Transmit / Receive
 Documents, Pages, Lines, Bits

5. Kingdom Time.

In any realtime system time is essential by definition and then we need a model for representing time. The CAN specification does not suggest any time model so one had to be created for the CAN Kingdom. The common model used in daily life is most often thought about as a linear one with year, month, day, hour and minute and no start nor end, just one reference, the birth of Christ. But in practice we often use a circular one, the wrist watch: It has a granularity of one second, twelve segments of five seconds, one revolution makes a minute, 60 revolutions makes a small circle, an hour, and 12 hours makes a big circle that completes the possibility to express time. A twelve-hour watch is good enough to represent time for most of us to carry out our daily work. In fact, within machines a circular model is most often the best choice to represent time.

There are two major classes of realtime systems: Event driven and Time driven. Over the years there have been discussions about which class is best. In society we mix both classes without thinking about it. For example, think about people transport systems: Underground represents a time driven system and taxi an event driven system. If you want to use the underground to get to a meeting you have to check the schedule and be at the station on time for the train departure. Alternatively you can call for a taxi and you will have it as soon as there is one available. Which alternative is the best one? There is no definite answer.

A Kingdom can belong to any or both of the classes of realtime systems and CAN Kingdom provides tools for the Kingdom Founder to arrange "Underground-type" as well as "Taxi-type" systems or to mix

them. For time driven systems we need one or more "Global Clocks" to provide a common time base within the system. One way to do this is to synchronize the local clocks to each other or to a master clock. The tools for this are King's Page 11, "Circular Time Base Setup Page" (page 73) and "Time Herald Document" (page 95). The King's Page 12, "Repetition Rate and Open Window Setup Page" (page 74) is used for scheduling. For event driven systems the King's Page 5, "Action/Reaction Setup Page" (page 62), can be used to link events to each other.

5.1 Circular Time.

Most of us are using a linear model of time. We think about it as an infinite line and relate all events to one specific point at that line, e.g.. the birth of Christ or Mohammed's flight to Medina. It has not always been like that. In the dawn of cultures the main need for measuring time was to predict the seasons and the regular behavior of rivers. Then the optimum time for sowing could be found. The first step was to find and establish the vernal equinox and the number of days to the next one to create a circular time model. When this was made a people had a model with 365 days per revolution, automatically calibrated each vernal equinox. A resolution of one day sufficient for everyday farming life. The next year would pass in the same way as the past year. Everything would repeat itself and there were no need for more future or past than could be expressed in days within a circle of one year. In the beginning was Circular Time.

When a culture grows more complex and the society involves more people then a need for better time measurements arises. Any state is based on laws and taxes and needs an administration to see to it that taxes are properly paid and that laws are obeyed. This requires power and dated files, kept over several years, to sustain the decrees. The art of war requires planning and actions based upon several years on the strategic level and synchronized behavior in the battle. Even in a more complex culture a circular model of time would do. The need for a description of time as an infinite time line does not arise until very abstract things as evolutionary and religious matters are to be discussed and even then the linear model is not evident. (The Maya culture used a time circle closed every 5126 year. Next time AD 2012 !) For most situations you will find a suitable circular model and resolution that will fit your need. The higher up in the hierarchy you are placed the bigger circle and rougher resolution you need to carry out your work for the society. For individuals at the bottom of the hierarchy, e.g.. a slave, a circular time of one day per revolution and one tenth of a second would do for any situation. This resolution comes from the fastest reaction a human body can show and within a day somebody at a higher level in society will tell the slave what to do.

From the discussion above we may conclude that a circular time will do for any machine and the only place where linear time may be required is at Man Machine Interface. The circumferential length and the resolution of a required time circle are depending on application. For a simple servo a circle of a tenth of a second and a resolution of some ten microseconds may be relevant but a furnace controller may require a circle of one day and a resolution in minutes might do. CAN Kingdom provides a Page to let the King set up one or more circular time bases in a Kingdom. The maximum time that can be expressed is 255 revolutions of a time circle and this circle can be divided into maximum 255 segments, each with a granularity of maximum 255 grains. The size of a grain can be set to 255 multiples of 10 ns in power of 0 - 9 (King's Page 11, page 73). This implies that the maximum time that can be expressed related to a specific instant is 33.554.430 grains, 16.777.215 grains representing past and 16.777.215 grains representing future. A hypothetical City with a time granularity of 1 ns would then have to have orders and to transmit reports within 16.7 ms to a supervising City in order to be time consistent. For a more

realistic City with a time granularity of $1 \mu\text{s}$ this period is 16.7 s. It is hard to keep clocks synchronized better than within $1 \mu\text{s}$ between Cities in a CAN Kingdom. The maximum time circle for a City that keeps a granularity of 1 second is 194 days. Some methods to adjust local clocks to master time are described in ref. [8].

The circular time model used in CAN Kingdom can be visualized as time revolving on an endless helical line round the surface of a torus (**Figure 7**).

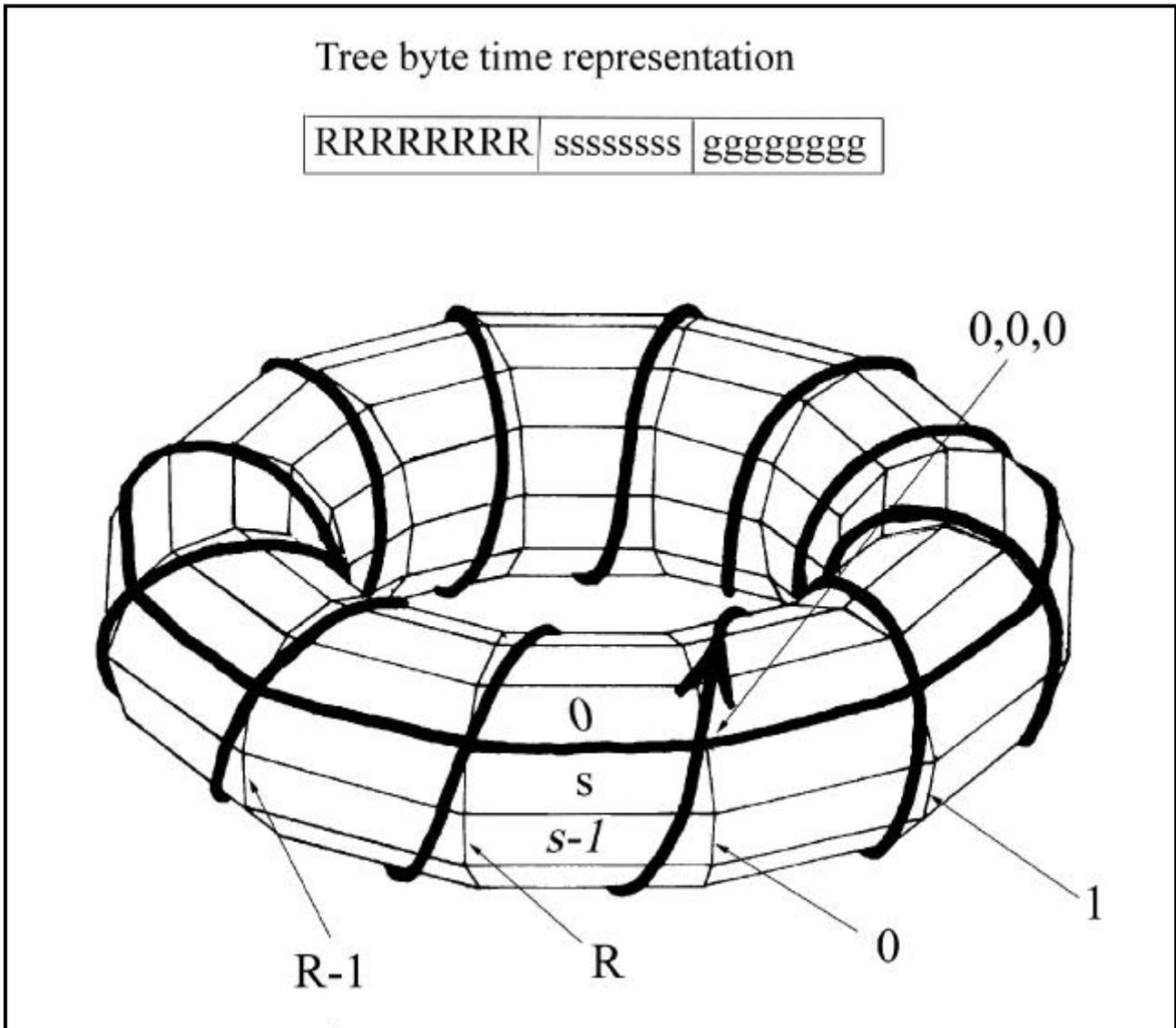


Figure 7. Circular Time model.

An example:

We would like to set up a time base like an ordinary watch with minutes and, hours presented and 12 hours as maximum time. A filled in Form for King's Page 8 may read as the one below:

Document name:	King's Document
Document List:	T1
Document Number:	0
Document type:	Transmit
<i>Page description.</i>	
Page number:	11
Number of Lines:	8
Data description:	The King's Page 8. Circular Time Base setting.
<p>Line 0: City or Group address</p> <p>Line 1: 00001011 (Page 11)</p> <p>Line 2: 00001001 10^9 ns Time granularity exponent. (1 s)</p> <p>Line 3: 00000001 Grain mantissa. (1 s)</p> <p>Line 4: 00111100 Number of grains per segment. (60 -> 1 min / segment)</p> <p>Line 5: 00111100 Number of segments per revolution. (60 min / rev. 1 rev. = 1 hour)</p> <p>Line 6: 00001100 Number of revolutions for maximum time. (12 hours)</p> <p>Line 7: rrrrrrr reserved.</p>	

5.2 Linear Time.

Within a machine a circular time expression will do but to communicate with the outer world we might need to shift to a linear model. In CAN Kingdom this can be done by coupling linear time to the local circular time on an appropriate level that makes the task simple (e.g., on a millisecond, second, minute or hour base) depending on the local granularity and communication bandwidth. The document for this is the Linear Time Document (paragraph 8.5, page 96).

5.3 Synchronization.

There are several ways to synchronize the different Cities in a Kingdom to each other. No specific way is prescribed in CAN Kingdom as the best method is depending on system requirements. In some Kingdoms it is quite OK to have one City responsible for a reference Kingdom Time, in others it may be unacceptable to rely on one City always working flawlessly. In some systems local clocks can be

resynchronized often but in other systems they may have to run accurately for a long period before they can be resynchronized. A Kingdom Founder should be able to use the most efficient procedure and a City designer can offer him many options by designing his City to support Action/Reaction Letters and/or time Letters. In this chapter only two hints will be given on how to synchronize Cities to each other.

5.3.1 A simple synchronization.

A simple and robust way to synchronize Cities is to relay on unadjusted local clocks with a relatively small time circle. The local clock of every City is set up alike and the Mayors are ordered to send an empty Letter when their respective local clock reaches the maximum time. They will all use the same Envelope. As soon as they receive this specific Null Letter, they reset their clocks. Such a behavior can be set up by Action/Reaction Documents. To support such a synchronization procedure a City will have the following Documents available:

- 1 Reaction Document: Clock reset.
This Document resets the local clock and stays then inactive for a period of x ms.
- 2 Support for King's Page 12, Repetition Rate and Open Window Setup Page.
- 3 A Null Document.

5.3.2 An example on a scheme for synchronizing a Kingdom where every City has an NEC 72005 Post Office.

The NEC 72005 circuit offers the possibility of getting an interrupt at each start bit until a Letter is received that is stored in buffer 0. The Time Herald Letter is allocated buffer 0. When a Time Herald Letter is received reading 1 on Line 0, the Mayor activates the start bit interrupts. At each start bit he reads the clock. When the Time Herald transmits the second Time Herald Letter reading 0 on Line 0 he reads his clock when the start bit of this message shows up. At all other Cities the value of their local clocks will be kept as the start bit interrupt procedure will stop when the Time Herald Letter is received correctly. If the Letter is flawlessly transmitted, the Time Herald writes 2 on Line 0 and the read clock value on Line 1 through Line 3 and sends this as the third Time Herald Letter. All receiving Mayors will correct their clocks with the difference between their reading of the local clock and the received "official time." When the described procedure is repeated a second time, the Mayors can detect if their individual local clock runs slower or faster than the clock of the Time Herald and adjust it to match the pace of the Time Herald clock. (ref. [8])

6. Bus Management.

The basic bus access management method of CAN is of a multiple access, arbitration by message priority type. This bus access management is implemented in silicon and thus always there. However, it is possible to use CAN as a base for higher level protocols which use other methods of carrying out bus

access management that give better real-time performance, e.g., a time driven access or a polling bus-master. The "standard" CAN procedure will then work as an emergency procedure that sees to it that the communication will not break down completely but resolve an accidental collision. I will give some examples on bus access management procedures for higher layer CAN protocols that will enhance real-time performance.

6.1 Event driven types.

For real-time systems it is important that the maximum latency time of at least some Letters can be predicted. The latency time is here defined as the time elapsed from an instant when the TX bit is set on the transmitting Post Office to the instant when the RX bit of the receiving Post Office is set. It is often said that only the highest priority Letter has a predictable latency time. This is true if no measures are taken to restrict the bus access. If the highest priority Letter is transmitted continuously, then no other Letter will be sent. Lower priority Letters will always lose arbitration. But why should a Letter continuously be retransmitted? It always takes some time to prepare a Letter to be transmitted and it always takes some time to read and act upon a received Letter. As the Postal System sees to it that a Letter is correctly transmitted then a Mayor can be restricted to reuse an Envelope for some time after it is used without any inconvenience. By such a restriction, resolving any queue and calculating the maximum latency time for any Letter is theoretically possible. An elaborate theory on how to calculate the proper amount of time to be elapsing before an Envelope is released for reuse can be found in ref. [9]. Here I will only show a simple example to explain the principle. First we organize Letters into groups:

1. The Envelopes 1 - 40 are divided into eight groups, numbered from 1 to 8, each containing five consecutive Envelopes.
2. After an Envelope in group N is transmitted, it must not be reused before $(3 \times N)$ ms has elapsed.
3. All other Envelopes may be used at any time.

Anticipate the following situation: It takes the Postman 200 μ s to deliver any Letter and the Mayors obey the rules above. Any Mayor is entitled to send any of his Letters. At a certain instant in time, all Mayors start mailing all their Letters at a maximum rate. The Postman will then deliver Letters according to the list below. Queued Letter groups are placed between brackets and when the Postman takes them, they appear in a bold typeface.

ms	Group	Envelopes of passed Letters
0 - 3	123(45678+ in the queue)	1 - 15
- 6	145	1 - 5, 16 - 25
- 9	126	1 - 10, 26 - 30
- 12	137	1 - 5, 11 - 15, 31 - 35
- 15	124	1 - 10, 16 - 20
- 18	158	1 - 5, 21 - 30, 36 - 40
- 21	123(6)	
- 24	167	
- 27	124(8)	
- 30	138	
- 33	125	
- 36	1xx	1 - 5 + ten unscheduled
- 39	123(46)	
-	...	

If we add 200 μ s to cover the case when the Postman had accepted to deliver a Letter just before the start of the sequence, we will have the following result:

Envelope no.	Max. latency time	Min. repeat time
1	400 μ s	3 ms
5	1.2 ms	3 ms
--	----	----
25	6.2 ms	15 ms
--	----	----
40	18.2 ms	24 ms
41	34.4 ms	0 ms

This shows that, in this case, an updating periodicity, regular or irregular, ranging from 0 Hz up to 333 Hz can be chosen for the five most frequent Letters and that the 41 highest prioritized Letters will have a predictable maximum latency time. As can be seen in the lists above we will have two free slots where unscheduled Letters have access to the bus and then the Letter with Envelope 41 will have a maximum latency time, although it has no time restriction at all. It may be continuously sent and still not hurt the transmissions of lower numbered Letters. If we like we can put restrictions on the Envelopes 42 - 50 and then the fifty highest priority messages will have a predictable latency time.

Keep in mind that no Letter has to be sent periodically! When the restriction time has elapsed for a certain Envelope, a Mayor may be free to use this Envelope whenever he likes. King's Page 10 (page 71) is used to set up time restrictions on Folder Labels. The method is described in ref. [10].

To implement the function of King's Page 10 the City needs a local clock. The King's Page 11 (page 73) allows the King to set the time base of a City but a City Founder may choose to implement a fixed setting. Then the parameters of King's Page 12 are filled-in in the City specification and the Form of King's Page 12 is marked "fixed" in the Page Form List.

6.2 Daisy chain types.

In a daisy chain type of bus access management one Letter from one City is used to trigger the Mayor of a second City to transmit a Letter that in turn triggers a third Mayor to do the same and so on. This procedure can be open or closed. In the open case the first Mayor transmits his Letter periodically or on a special event. In the closed case the Letter from the last Mayor triggers the first Mayor to start a second round. King's Page 5 (page 67) can be used for setting up a daisy chain procedure.

Example 1. A closed daisy chain:

Assume a small Kingdom with five temperature sensor Cities and five pressure sensor Cities. The King distributes the following transmit and receive Envelopes:

City	Envelopes	
	Transmit/Folder	Receive/Folder
Temp. 1	10/3	51/4
Temp. 2	20/3	10/4
Temp. 3	30/3	20/4
Temp. 4	40/3	30/4
Temp. 5	50/3	40/4
Press. 1	11/3	51/4
Press. 2	21/3	11/4
Press. 3	31/3	21/4
Press. 4	41/3	31/4
Press. 5	51/3	41/4

Then the King sends a King's Page 5 that reads:

Document name:	King's Document	
Document List:	T1 Capital / 0 City	
Document Number:	0 Capital / 0 City	
Document type:	Transmit (Capital) Receive (City)	
<i>Page description.</i>		
Page number:	5	
Number of Lines:	8	
Data description:	The King's Page 5. Action Page - Reaction Page	
Line 0:	00000000	Group Number All Cities.
Line 1:	00000101	Page 5
Line 2:	00000100	Folder Number 4 Folder containing the Acting Document to which the Mayor must react.
Line 3:	00000000	Form Number The Form of the Acting Document to which the Mayor must react.
Line 4:	rrrrrrrr	r = 0 Reserved.
Line 5:	00000011	Folder Number 3 Folder containing the Reacting Document according to which the Mayor shall respond.
Line 6:	00000000	Form Number The Form of the Reacting Document according to which the Mayor shall respond.
Line 7:	rrrrrrrr	r = 0 Reserved.

All Mayors will receive the King's Letter with one transmission. By sending a Remote Request of Letter 51 (or any other one Letter) the King then initiates the process. As the lowest priority Letters (50 and 51) initiates the highest priority Letters (10 and 11) then every Mayor will have access to the bus for one Letter per revolution. Here temperature values and pressure values may come mixed in different ways, mutually in consecutive orders. If Temp.1 is set to trigger Press.1 that in turn triggers Temp.2 and so on, we will interlace temperature and pressure Letters. Obviously other schemes are possible as well.

Example 2. An open daisy chain.

The procedure above has only to be slightly adjusted to create an open daisy chain. The Envelope distribution is made as follows:

City	Envelopes	
	Transmit/Folder	Receive/Folder
Temp. 1	10/3	9/4
Temp. 2	20/3	10/4
Temp. 3	30/3	20/4
Temp. 4	40/3	30/4
Temp. 5	50/3	40/4
Press. 1	11/3	9/4
Press. 2	21/3	11/4
Press. 3	31/3	21/4
Press. 4	41/3	31/4
Press. 5	51/3	41/4

The Letter 5 is the same as in Example 1. Now the King can trigger the chain by transmitting an empty Letter 9. The chain will run once every time Letter 9 is transmitted.

Example 3. A Bus-Master approach.

A Bus-Master access management can be achieved by setting up the following Envelope distribution:

City	Envelopes	
	Transmit/Folder	Receive/Folder
Temp. 1	10/3	8/4
Temp. 2	20/3	8/4
Temp. 3	30/3	8/4
Temp. 4	40/3	8/4
Temp. 5	50/3	8/4
Press. 1	11/3	9/4
Press. 2	21/3	9/4
Press. 3	31/3	9/4
Press. 4	41/3	9/4
Press. 5	51/3	9/4

In this example all Pressure City Mayors will transmit their measured pressure whenever the Bus-Master (not necessarily the King) will transmit Letter 8 and all Temperature City Mayors whenever Letter 9 is sent. The Bus-Master can in this way release the bus for individual Letters or groups of Letters.

6.3 Time driven types.

Time driven systems require a global clock, i.e., time in any City of the Kingdom is related to a Kingdom wide time reference. Such a feature can be implemented in many ways and CAN Kingdom only assumes

that each City has a local clock synchronized to a global clock at certain intervals. A local clock might also be adjusted to run closely at the same pace as the global clock. How well the local clocks have to be synchronized to the global clock depends entirely on the system requirement and architecture.

CAN Kingdom provides the following tools to set up a time driven protocol:

1. King's Page 11, Circular Time Base Setup Page.
2. King's Page 12, Repetition Rate and Open Window Setup Page.
3. Time Herald Document.

By King's page 11 a proper time base can be set up at each City and the local clock can be synchronized by Time Herald Pages sent at appropriate points in time. This is straight forward but the use of King's Page 12 might need some explanation. You find the Form for King's Page 12 on page 74. The repetition rate of actions initiated by a Document, e.g., the transmission of one Letter, is controlled by the Folder where it is placed. Line 2 on the King's Page 9 tells the Mayor of the City which of his Folders to be set up this time. The Mayor starts up the repetition handling procedure whenever his local time is zero. When the time reaches the value of Line 3 and 4 (Start Segment offset and Start Revolution offset) the action(s) of the Document in the Folder is released for the first time. By the Line 5 the King can limit the time for this action. If the action is to transmit a Letter then the Mayor orders his Postmaster to transmit it at the time of Line 3 and 4. Maybe the priority of the Letter is low and the Postmaster could never hand it over to the Postman during the time of "Open window." If this is the case, then the Mayor cancels the transmission order and the Letter is not sent this time. The next time possible is given by Line 6 and 7 and a new attempt is made to send the (updated) Letter. The procedure is always started at time zero, the action released for the first time given on line 4 and 5 and repeated at an interval given by line 6 and 7 until time is back to zero.

In this way the Kingdom Founder can control the communication. Important Letters can be sent in time slots during which no other Letters are sent. Less important Letters can be sent in time slots where they will be sorted according to priority. Other actions, like reading sensors, can be ordered to take place at specific points in time by King's Page 12 as well. Simultaneous readings of sensors can often reduce the needed bandwidth of the communication.

7. The Block.

Most Documents contain information that can simply be arranged into Forms in a meaningful way but occasionally this is not so easy. One Mayor may need to have an executable code from another one or wants to send a larger amount of data organized in another format than eight bit bytes. In order to fit into a Page (CAN Data Frame), such data have to be packed into a suitable format, e.g., ASCII code or Intel-Hex, before transmission and unpacked and restored to its original pattern at reception. To handle such data we introduce the concept **Block**.

During a transmission of a Block, the information will be temporarily organized into Forms of **Block Transfer Documents**. The Mayor of any CAN Kingdom City, who will transmit or receive a Block, have Block Transfer Documents as a tool to pack a bit stream for transmission by the Postal System and, at reception, to unpack the bits and restore the bit information into its original shape. The shape is depending on the type of information in question and it is the task of the Kingdom Founder to see to it that the transmitting and receiving Cities are compatible.

The transmission of Blocks is controlled by the King in the same way as he controls all other communication: He prepares the possibility for the transmission by assigning an Envelope to Folders, each containing a Block Transfer Document containing the Block. Then the Mayors take care of the transmission on their own. The King himself will execute a Block transfer in the same way as any Mayor.

The capacity of a receiving City to unpack Blocks and restore them into the original form varies depending on different software approaches and on other tasks that a Mayor may have to fulfil at the same time. This requires a tool for the receiver to set the pace of incoming Block Transfer Pages and this tool is the Bundle Request. A Bundle is a set of consecutive Pages and the receiving Mayor instructs the transmitting Mayor to send one Bundle and then wait for an acknowledgment before transmitting the next Bundle.

7.1 The Block Transfer Document.

To execute a Document containing a Block, each Mayor needs two copies of the Block Transfer Documents: One in a Transmit Folder and another one in a Receive Folder. By this arrangement we have also a possibility to have a half duplex Block communication channel between Mayors if needed.

The Block Transfer Documents is outlined in the same way as the King's Document: All Pages contain eight Lines where the second line is a pagination. As the City or Group address is useful only for the King, Line zero is kept in reserve on the Block Transfer Documents Forms for future use.

A Block Transfer Document contains Forms for the following five Pages:

- Page 0. Request Page. (Only sent by the receiver. Optional.)
The receiving Mayor requests the transmission of a Block.
- Page 1. Setup Page. (Only sent by the transmitter.)

The transmitter informs the receiving Mayor about the length of the Block to be transmitted.

- Page 2. Bundle Request Page. (Only sent by the receiver.)
The receiving Mayor informs the transmitter about how many consecutive Information Transport Pages he wants to receive at a time. A successful session is terminated by the receiver requesting zero pages.
- Page 3 & 4. Information Transport Page. (Only sent by the transmitter.)
A Block is divided into six byte packages and each package is transmitted by a Letter containing by turn Page 3 or 4. A Page 3 is always the first Page in a Bundle.
- Page 5. Abort Page. Terminates the Block transfer session on error.

The Block Transfer Document Page Forms are specified in paragraph 8.3, page 89.

7.2 Block Transfer Procedure.

The exchange of Blocks between Cities is organized by the King in the same way as exchange of any other Documents. The King opens a communication channel for the transmission of a specific Block by assigning proper Envelopes to the Folders containing the Block Transfer Documents at each City. Then the exchange can take place on the initiative of either the receiving or the transmitting Mayor. If the receiver takes the initiative, the procedure is the following:

- Step 1. The receiving Mayor sends a Request Page, Page 0.
- Step 2. The transmitting Mayor answers by sending a Setup Page, Page 1, telling the receiving Mayor the total number of Lines (eight bit bytes) he will receive during this session.
- Step 3. The receiving Mayor sends a Bundle Request Page, Page 2, telling the transmitting Mayor how many consecutive Information Transport Pages (Page 3 or 4) he must send before expecting a new Bundle Request Page.
- Step 4. The transmitting Mayor sends a Bundle of Information Transport Pages, Page 3 and 4 alternatively, starting with Page 3. By this toggling procedure the receiver can detect a double transmission and discard one of them¹. If the Bundle Request Page is not acceptable (which indicates a fatal error), an Abort Page, Page 5, is sent instead and the procedure is terminated.

¹A receiver monitoring a dominant bit at the last bit of End of Frame will nevertheless accept the message as correct but a transmitter, monitoring a dominant bit at the same place, will retransmit the message. This will result in a duplicate message transmission, quite in accordance with the CAN specification.

- Step 5. The receiver sends a new Bundle Request Page when he is prepared for a new Bundle of Page 3:s and 4:s. If the latest Bundle is not accepted, an Abort Page (Page 5) is sent instead and the procedure is terminated.
- Step 6. Step 4 and 5 are repeated until the transmitter has sent the number of Lines according to the Setup Page.
- Step 7. When the receiving Mayor has received the last Bundle of Information Transport Pages, he checks that he has received the right number of Lines. If so he terminates the session by sending a Bundle Request Page requesting no pages. If not so, he sends an Abort Page.

If the transmitting Mayor takes the initiative, step 1 is omitted.

The Block Transfer Documents are organized as follows to create an information channel:

Transmitting City	Receiving City	Page Type
Transmit Document	Receive Document	
Page 1	Page 1	Setup Page.
Page 3	Page 3	Information Transport Page.
Page 4	Page 4	Information Transport Page.
Page 5	Page 5	Abort Page.
Receive Document	Transmit Document	
Page 2	Page 2	Bundle Request Page.
Page 5	Page 5	Abort Page.

8. CAN Kingdom Documents.

CAN Kingdom has four kinds of standard Documents: The **King's Document**, the **Mayor's Document**, the **Block Transfer Document** and the **Time Herald Document**. These Documents are used by the Kingdom Founder to establish rules for the Kingdom and to provide the King with means to rule the Kingdom. Compulsory for the Mayor of a City is to have at least the Mayor's Document (Page 0 and 1 for identification of the City) and the King's Document. The King's Document has at least three Pages

(0, 1, and 2) but may have more Pages depending on the flexibility of the City. In the paragraphs below the Page Forms of the Documents are presented. It may be worth mentioning that the Page Forms only reflects different services that a Mayor offers the King, not how these services are implemented. The coding can be made in several ways, some more efficient than others.

8.1 The King's Document.

The King's Document is the most important tool for the CAN Kingdom Founder. With this he can make the Kingdom efficient. At the setup sequence the organization of the Kingdom can be checked and adjusted. He also can configure general Cities according to his actual need. For system dependent Cities he can let them know to which Kingdom they belong by telling their Mayors who the King is.

The King's Letters make it possible for the City Founder to design a City that can be fitted to different Kingdoms without any physical adjustments before or after it is connected to the system. Any change needed is made by software adjustments. These are initiated by different Pages in the King's Document, sent from the Capital as King's Letters. A King's Letter consists of the King's Envelope and one of the King's Pages.

The King's Document can have up to 256 different King's Pages (and thus 256 different King's Letters). The first thirty two Pages (0 to 31) are reserved for general system Pages. The next ninety-six Pages (32 to 127) are reserved for branch organizations, such as "The Can Textile Users Group" or "The CAN Hydraulic Users Group," and will be specified for branch specific needs. The last 128 Pages can be used by City Founders at will.

The Pages in the King's Document are organized as follows:

Page 0	Start/Stop, Modes	(Compulsory)
Page 1	Initiate (Base Number,Response request Page)	(Compulsory)
Page 2	Assign Envelopes	(Compulsory)
Page 3	Assign Groups	(Optional)
Page 4	Remove Groups	(Optional)
Page 5	Action Page - Reaction Page.	(Optional)
Page 6	Reserved.	
Page 7	Reserved.	
Page 8	Bit timing registers setting.	(Optional)
Page 9	New City physical address.	(Optional)
Page 10	Time Elapse.	(Optional)
Page 11	Circular Time Base Setup Page.	(Optional)
Page 12	Repetition Rate and Open Window Setup Page.	(Optional)
Page 13 - 15	Reserved	
Page 16	Place Documents into Folders.	(Optional)
Page 17	Create Documents, Forms or Lines from Lists.	(Optional)

Page 18	Create Compressed Letters.	(Optional)
Page 19	First Identifier Masks.	(Optional)
Page 20	Create Filters for a Postmaster.	(Optional)
Page 21 - 31	Reserved	

The City address is always the same as their Physical number (n) in the Postal System. In addition all Cities have a common address 0, through which they can receive broadcasted King's Letters.

8.1.1 The **King's Page 0**. (Compulsory) Terminates the setup phase and sets a City in appropriate modes.

The King's Page 0 is used to tell the Mayors that the setup sequence is completed and that they should bring their respective City into work. It is also used during the run phase to freeze the activity of a City or a Group of Cities, stop transmitting any Letters, reset, etc. When a City is in **freeze** mode, its Mayor must be able to handle all implemented types of the King's Letter. A City can have more than one run, freeze and/or reset mode which can be selected with Page 0. The King can address this letter to a single City, a Group of Cities or all the Cities.

Form for the King's Page 0:

Document name:	King's Document	
Document List:	T1 Capital / 0 City	
Document Number:	0 Capital / 0 City	
Document type:	Transmit (Capital) Receive (City)	
<i>Page description.</i>		
Page number:	0	
Number of Lines:	8	
Data description:	The King's Page 0. Terminates the setup phase. Orders a Mayor to set his City into a specific working mode, e.g., in a Run or Freeze mode.	
<i>Line description.</i>		
Line 0:	City or Group address	
Line 1:	00000000 (Page 0)	
Line 2:	rrrrrAA	Action Mode AA = 00 Keep current mode AA = 01 Run AA = 10 Freeze AA = 11 Reset r = 0 Reserved
Line 3:	rrrrrCC	Communication Mode CC = 00 Keep current mode CC = 01 Silent CC = 10 Listen Only CC = 11 Communicate r = 0 Reserved
Line 4:	MMMMMMMM	City Mode M = 0 Keep current Mode. M ≠ 0 Modes according to the City specification.
Line 5:	rrrrrrr	r = 0 Reserved
Line 6:	rrrrrrr	r = 0 Reserved
Line 7:	rrrrrrr	r = 0 Reserved

Action Mode: Relates to actual City Mode and Communication Mode. Action Modes have to be defined by the City Founder.

Communication Modes:

Silent: The Postmaster will be silent but still receive Letters and notice the Mayor when Letters are accepted. The Postmaster will not transmit any acknowledgment bit nor error or over load frame.

Listen Only: The Postmaster will be fully active but the Mayor will send Letters only upon the King's request.

Communicate: Normal communication.

City Mode: City specific modes, e.g., configuration mode, service mode, working mode, etc. For each City Mode the City Founder has to define how the City will work on Action Mode and Communication Mode commands from the King.

8.1.2 The **King's Page 1**. (Compulsory) Initiating Page. Provides the Base Number and asks for Mayor's response.

By this Letter the King can set or change the Base Number that is to be used for the Mayor's Document (see page 87). The Mayor will execute the change immediately and respond according to the request.

During the run phase this Letter can be used to check the entire Kingdom or certain Groups, e.g., check up the Mayors whether they are awake or not.

Form for the King's Page 1:

Document name:	King's Document	
Document List:	T1 Capital / 0 City	
Document Number:	0 Capital / 0 City	
Document type:	Transmit (Capital) Receive (City)	
<i>Page description.</i>		
Page number:	1	
Number of Lines:	8	
Data description:	The King's Page 1. Initiating Page. Provides the Base Number and asks for Mayor's response..	
<i>Line description.</i>		
Line 0:	City or Group address	
Line 1:	00000001	(Page 1)
Line 2:	xxxxxxx 11111111	The Mayor's response Page in Mayor's Document. No response requested.
Line 3:	r r r r r r r r	r = 0 Reserved
Line 4:	xxxxxxx	Base Number, LSB
Line 5:	xxxxxxx	Base Number
Line 6:	xxxxxxx	Base Number
Line 7:	Errxxxxx	Base Number, MSB
		All x = 0 -> Keep current setting. All x = 1 -> Base Number undefined.
	rr = 00	Reserved
	E = 1/0	Extended / Std. format.

When a Mayor receives this Page he will, as soon as possible, respond with the requested Page in his Mayor's Document.

8.1.3 The **King's Page 2.** (Compulsory) Assigning an Envelope to or expelling an Envelope from a Folder forming Letters and restraining the use of an Envelope.

Each Mayor in the Kingdom has always an appropriate set of Pages from the Kings Document placed in Folder 0 and a Mayor's Document in Folder 1. The Envelope 0 is assigned to the former Folder. As

soon as a Mayor is informed about the Base Number, he will assign the proper Envelope (Base Number + n) to the latter Folder. These are the only two Letters that he can use so far. For all other Letters the King has to allot Envelopes. The tool for this is Page 2 of the King's Document.

During run phase the King can change Envelopes in order to change the priority of transmissions (ref. [10]). If all Mayors in a Group use the same Folder for corresponding Documents, the King may easily switch an information flow on or off by enabling/disabling the use of an Envelope, just by multicasting King's Page 2.

Important: The City Founder has to specify how much time it will take to execute the tasks initiated by King's Letter, Page 2, and how the City will work during this period!!

Important: A specific Envelope can be set as *transmit* on one and *only* one of the enabled Folders within the entire Kingdom, except when the Folder contains a Null Document.



Form for the King's Page 2:

Document name:	King's Document	
Document List:	T1 Capital / 0 City	
Document Number:	0 Capital / 0 City	
Document type:	Transmit (Capital) Receive (City)	
<i>Page description.</i>		
Page number:	2	
Number of Lines:	8	
Data description:	The King's Page 2. Assigning an Envelope to or expelling an Envelope from a Folder forming Letters and restraining the use of an Envelope.	
<i>Line description.</i>		
Line 0:	City or Group address	
Line 1:	00000010	(Page 2)
Line 2:	xxxxxxx	Envelope Number, LSB
Line 3:	xxxxxxx	Envelope Number
Line 4:	xxxxxxx	Envelope Number
Line 5:	ECrxxxx	Envelope Number, MSB
	r = 0	Reserved
	C = 1/0	Compressed Envelope yes/no.
	E = 1/0	Extended / Std. format.
Line 6:	xxxxxxx	Folder Number
Line 7:	rrrrAAE	E = 1/0 Enable/Disable the use of this Envelope. (Folder Number is ignored.)
	AA = 00	Keep current assignment. (Folder Number is ignored.)
	AA = 01	Assign this Envelope to the Folder on Line 6. If another Envelope is already assigned to this Folder, the previous assignment will be kept in parallel. The use of this Envelope is enabled or disabled according to the value of E.
	AA = 11	Transfer the current assignment of this Envelope to the Folder on Line 6. The old assignment is canceled. The use of this Envelope is enabled or disabled according to the value of E.
	AAE = 100	Expel this Envelope from any assignment. (Folder Number is ignored.)
	rrrr = 00000	Reserved
		All other combinations are reserved.

8.1.4 The **King's Page 3**. (Optional) Assigning a City to Groups.

All Cities in the Kingdom belong to a Group with the address **0**. This address can be used for broadcasting purposes. Assigning a City to additional Groups is also possible. The Group address is given by the King and can be any number between 1 and 255 not used as a City address. This feature can, e.g., be used to freeze a part of the Kingdom.

Form for the King's Page 3:

Document name:	King's Document	
Document List:	T1 Capital / 0 City	
Document Number:	0 Capital / 0 City	
Document type:	Transmit (Capital) Receive (City)	
<i>Page description.</i>		
Page number:	3	
Number of Lines:	8	
Data description:	The King's Page 3. Assigning a City to Groups.	
<i>Line description.</i>		
Line 0:	City address	
Line 1:	00000011	(Page 3)
Line 2:	xxxxxxx	Address number of the first Group to which the City is assigned.
Line 3:	xxxxxxx	Address number of the second Group.
	00000000	No more assignments.
Line 4:	Equivalent to Line 3.	
Line 5:	Equivalent to Line 3.	
Line 6:	Equivalent to Line 3.	
Line 7:	Equivalent to Line 3.	

If the City will belong to more than six Groups, additional King's Pages number 3 are sent.

8.1.5 The **King's Page 4**. (Optional) Removing a City from Groups.

Form for the King's Page 4:

Document name:	King's Document	
Document List:	T1 Capital / 0 City	
Document Number:	0 Capital / 0 City	
Document type:	Transmit (Capital) Receive (City)	
<i>Page description.</i>		
Page number:	4	
Number of Lines:	8	
Data description:	The King's Page 3. Remove a City from Groups.	
<i>Line description.</i>		
Line 0:	City address	
Line 1:	00000100	(Page 4)
Line 2:	xxxxxxx	Address number of the first Group from which the City is removed.
Line 3:	xxxxxxx 00000000	Address number of the second Group. No more removals. (Every City will always belong to Group 0.)
Line 4:	Equivalent to Line 3.	
Line 5:	Equivalent to Line 3.	
Line 6:	Equivalent to Line 3.	
Line 7:	Equivalent to Line 3.	

8.1.6 The **King's Page 5**. (Optional) Action/Reaction Setup Page.

By this Page the King can instruct a Mayor to let a Page in one Document react on another Page in another Document. This feature can be used for application acknowledgment, to create different types of message scheduling, etc. The function can be controlled by enabling/disabling the folders and the link can be deleted by removing a Document from the actual Folder.

Form for the King's Page 5:

Document name:	King's Document	
Document List:	T1 Capital / 0 City	
Document Number:	0 Capital / 0 City	
Document type:	Transmit (Capital) Receive (City)	
<i>Page description.</i>		
Page number:	5	
Number of Lines:	8	
Data description:	The King's Page 5. Action Page - Reaction Page	
<i>Line description.</i>		
Line 0:	City or Group address	
Line 1:	00000101	(Page 5)
Line 2:	FFFFFFF	Folder Number Folder containing the Acting Document to which the Mayor must react.
Line 3:	PPPPPPP	Form Number Form of the Page in the Acting Document to which the Mayor must react.
Line 4:	rrrrrrr	r = 0 Reserved.
Line 5:	fffffff	Folder Number Folder containing the Reacting Document by which the Mayor must respond.
Line 6:	ppppppp	Form Number Form of the Page in the Reacting Document by which the Mayor must respond.
Line 7:	rrrrrrr	r = 0 Reserved.

8.1.7 The **King's Page 8**. (Optional) Postmaster control register settings.

Postmaster control register settings as pre-scaler, number of bit timing quanta, sampling point, resynchronization jump width, number of sampling points, remote transmit mode, slope control, etc. differs between different makes of hardware. Therefore CAN Kingdom proposes no general standard procedure for these settings but a King's Page 8, specific to each CAN Controller. New Postmaster control registers will be valid after reception of a King's Page 0 ordering reset.

Form for the King's Page 8:

Document name:	King's Document	
Document List:	T1 Capital / 0 City	
Document Number:	0 Capital / 0 City	
Document type:	Transmit (Capital) Receive (City)	
<i>Page description.</i>		
Page number:	8	
Number of Lines:	8	
Data description:	The King's Page 8:0. New bit-timing register settings for Motorola 68HCx05Xx.	
<i>Line description.</i>		
Line 0:	City or Group address	
Line 1:	00001000	(Page 8)
Line 2:	00000000	CAN Controller type 68HCx05Xx.
Line 3:	rnnnnnnn	n Identification number of the CAN controller in the City. (City defined) r = 0 Reserved.
Line 4:	S ₁ S ₀ B ₅ B ₄ B ₃ B ₂ B ₁ B ₀	MCAN Bus Timing Register 0 (CBT0): S _x = SJW _x , B _x = BRP _x
Line 5:	ST ₂₂ T ₂₁ T ₂₀ T ₁₃ T ₁₂ T ₁₁ T ₁₀	MCAN Bus Timing Register 1 (CBT1): S = SAMP, T _x = TSEG _x
Line 6:	rrrrrrrr	
Line 7:	rrrrrrrr	r = 0 Reserved.

King's Page 8 Forms for other CAN Controllers:*Philips 82C200:*Line 2: 00000001_B CAN controller type Philips 82C200

Line 4, 5 : analogous to Motorola 68HC05Xx.

*Intel 82526:*Line 2: 00000010_B CAN controller type Intel 82526

Line 4, 5 : analogous to Motorola 68HC05Xx.

*Intel 82527:*Line 2: 00000011_B CAN controller type Intel 82527

Line 4, 5 : analogous to Motorola 68HC05Xx.

*NEC μ PD72005:*Line 2: 00000100_B CAN controller type NEC μ PD72005Line 4 : C00B₄B₃B₂B₁B₀ Contents of the BRPRS register.
C = CEBRPRS bit.
B_x = BRPRS_xLine 5: S₂S₁S₀D₄D₃D₂D₁D₀ Contents of the SYNC0 register.
S_x = SPT_x
D_x = DBT_xLine 6: 000YW₁W₀S₄S₃ Y = SYNC bit
W_x = SJW_x
S_x = SPT_x**For information about CAN Controllers not defined above, please contact KVASER AB.**

8.1.8 The **King's Page 9**. (Optional) Change of City physical address.

By this Page the King can instruct a Mayor to change the physical address of his City. The Mayor will execute the changing immediately and respond according to the request (if he has got the Base Number).

Document name:	King's Document	
Document List:	T1 Capital / 0 City	
Document Number:	0 Capital / 0 City	
Document type:	Transmit (Capital) Receive (City)	
<i>Page description.</i>		
Page number:	9	
Number of Lines:	8	
Data description:	The King's Page 9. New City Address.	
<i>Line description.</i>		
Line 0:	cccccccc	Current City address
Line 1:	00001001	(Page 9)
Line 2:	xxxxxxx 11111111	The Mayor's response Page in the Mayor's Document. No response requested.
Line 3:	xxxxxxx	New City Address.
Line 4:	rrrrrrrr	r = 0 Reserved.
Line 5:	rrrrrrrr	r = 0 Reserved.
Line 6:	rrrrrrrr	r = 0 Reserved.
Line 7:	rrrrrrrr	r = 0 Reserved.

8.1.9 **The King's Page 10.** (Optional) Minimum time elapsing between two consecutive transmissions of the same Envelope.

When the Postmaster has taken care of a Letter, he sees to it that it will be properly transmitted. When a Mayor receives a Letter, he always needs some time to read it and to act upon it. Therefore, once a Letter is sent, there will always elapse some time before a new Letter with the same Envelope is required. By allowing some time to elapse before the same Envelope is used again, organizing the flow of Letters in such a way that many Letters will have a predictable maximum latency time is possible. This is described in ref. [10].

King's Page 10 orders a Mayor to let a specified time elapse before he sends a new Letter with the same Envelope as a previous one.

Form for the King's Page 10:

Document name:	King's Document	
Document List:	T1 Capital / 0 City	
Document Number:	0 Capital / 0 City	
Document type:	Transmit (Capital) Receive (City)	
<i>Page description.</i>		
Page number:	10	
Number of Lines:	8	
Data description:	The King's Page 10. Minimum time elapsing between two consecutive transmissions of the same Envelope.	
<i>Line description.</i>		
Line 0:	City address	
Line 1:	00001011	(Page 10)
Line 2:	xxxxxxx	Envelope Number, LSB
Line 3:	xxxxxxx	Envelope Number
Line 4:	xxxxxxx	Envelope Number
Line 5:	ECrxxxx	Envelope Number, MSB
	r = 00	Reserved
	C = 1/0	Compressed Envelope yes/no.
	E = 1/0	Extended / Std. format.
Line 6:	sssssss	Number of segments.
Line 7:	rrrrrrr	Number of revolutions.
	Prohibited period between two consecutive transmissions is given by Line 6 and 7 according to the King's Page 11, Circular Time Base Setup Page.	

8.1.10 The **King's Page 11**. (Optional) Circular Time Base Setup Page.

The use of King's Page 11 is discussed in detail in paragraph 45, "Circular Time," page 45. Time begins at zero. A City may have fixed values on some or all lines in this Page.

Form for the King's Page 11:

Document name:	King's Document	
Document List:	T1 Capital / 0 City	
Document Number:	0 Capital / 0 City	
Document type:	Transmit (Capital) Receive (City)	
<i>Page description.</i>		
Page number:	11	
Number of Lines:	8	
Data description:	The King's Page 11. Circular Time Base setting.	
<i>Line description.</i>		
Line 0:	City or Group address	
Line 1:	00001011	(Page 11)
Line 2:	nnnnnnnn	10^n ns Time granularity exponent.
Line 3:	xxxxxxx	Grain mantissa.
Line 4:	nnnnnnnn	Number of grains per segment.
Line 5:	NNNNNNNN	Number of segments per revolution.
Line 6:	RRRRRRRR	Number of revolutions for maximum time.
Line 7:	r r r r r r r	r = 0 Reserved.

8.1.11 The **King's Page 12**. (Optional) Repetition Rate and Open Window Setup Page.

Form for the King's Page 12:

Document name:	King's Document	
Document List:	T1 Capital / 0 City	
Document Number:	0 Capital / 0 City	
Document type:	Transmit (Capital) Receive (City)	
<i>Page description.</i>		
Page number:	12	
Number of Lines:	8	
Data description:	The King's Page 12. Repetition rate and open window setting.	
<i>Line description.</i>		
Line 0:	City or Group address	
Line 1:	00001100	(Page 12)
Line 2:	FFFFFFFF	Folder Number.
Line 3:	$o_s o_s o_s o_s o_s o_s o_s o_s$	Start Segment offset.
Line 4:	$O_R O_R O_R O_R O_R O_R O_R O_R$	Start Revolution offset.
Line 5:	$n n n n n n n n$ $n = 0$	Open window. Number of segments. Window always open.
Line 6:	$i_s i_s i_s i_s i_s i_s i_s i_s$	Repetition increment (within a revolution), number of segments.
Line 7:	$i_R i_R i_R i_R i_R i_R i_R i_R$	Repetition increment, number of revolutions.

8.1.12 The **King's Page 16**. Setting the Folder Label and/or placing a Document into a Folder. Placing a Document into a Folder.

All Cities have predefined Documents from the beginning or constructed by the King through King's Letter, Page 17. Documents are placed into Folders in order to ensure safe handling throughout the Kingdom. Mayors, who have the Kings Page 16 implemented, make it possible for the Kingdom Founder to see to it that all Cities in a Group have matching Documents in Folders with the same Folder Number. Then a Group address can be used when shifting Envelopes or changing Folder information as enable / disable etc.

Line 3 on this Form contains the RTR bit and the six bits of the Control field specified in the CAN specification. A City Founder may allow the Kingdom Founder to manipulate these bits. If so it is important that the City will obey some rules:

1. The City documentation has to clearly show how each setting will affect received or transmitted Letters.
2. The DLC is according to the CAN specification and will appear on the CAN message as written on Line 3 and the data length is adjusted to fit the DLC.

The RTR bit as specified in the CAN specification is open for several interpretations. On top of that a Folder can contain a multi-Page Document. It has to be defined for each Transmit Document of a City how it will react upon a remote request, e.g., if all Pages will be sent consecutively or a specific Page will be sent, if a remote request will be ignored when the RTR bit on the Folder is not set, etc.

Form of the King's Page 16:

Document name:	King's Document
Document List:	T1 Capital / 0 City
Document Number:	0 Capital / 0 City
Document type:	Transmit (Capital) Receive (City)
<i>Page description.</i>	
Page number:	16
Number of Lines:	8
Data description:	The King's Page 16. Setting the Folder Label and/or placing a Document into a Folder.
<i>Line description.</i>	
Line 0:	City address
Line 1: 00010000	(Page 16)
Line 2: xxxxxxxx	Folder Number 2 - 255 (Folder 0 contains the Kings's Document and Folder 1 the City Identification Document.)
Line 3: 1Rrrx ₃ x ₂ x ₁ x ₀	x ₀ - x ₃ = DLC0 - DLC3 rr = 0 reserved bits R = RTR bit 00000000 Current settings of this Line remain unchanged.
Line 4: 1EZIr r r T	T = 1/0 Transmit/Receive. rrr = 000 Reserved. I = 1/0 Insert the Document on Line 5 and 6: Yes/No Z = 1/0 Remove the previous Document: Yes/No E = 1/0 Enable / disable the use of this Folder. ZI = 01 Illegal. A Folder can only contain one Document. 00000000 Current Line settings remain unchanged.
Line 5: xxxxxxxx	Document List Number, Transmit or Receive according to "T" on Line 4. Ignored if I=0.
Line 6: xxxxxxxx	Record number in the Document List. Ignored if I=0.
Line 7: rrrrrrrr	r = 0 Reserved

Line 3, note:

According to the CAN specification (ref. 1) a "Remote frame" must have the same DLC as the corresponding "Data frame" throughout the whole system.

Comments on Line 4:

With the Enable / Disable switch on Line 4 the Kingdom founder can switch on or off an information flow of a whole Document from or to a City or to Cities in a Group.

If a Transmit Document is placed in the Folder and bit T = 0 (Receive), this will be accepted but no transmissions will take place, regardless of the value of E, until T is set to 1.

If the Mayor is ordered to make any settings that seem wrong from his position he should send a Mayor's Letter to the King that informs him about this situation.

8.1.13 The **King's Page 17**. Create a Document, a Form or a Line from predefined Form-, Line- or Bit-Lists.

When a Postmaster receives a Letter he will have the Lines in the same order as they were sent. Then it is needed that the sending Mayor and the receiving Mayor have matching Letters and Pages. As the Kingdom Founder has chosen specific Cities for his Kingdom, he also knows what information each City needs, what information it can send and the data structure used. With the King's Page 17 the Kingdom Founder can create Documents, Forms and Lines (targets) from defined Page Form Lists, Line Form Lists and Bit Form Lists (sources).

Form for the King's Page 17:

Document name:	King's Document	
Document List:	T1 Capital / 0 City	
Document Number:	0 Capital / 0 City	
Document type:	Transmit (Capital) Receive (City)	
<i>Page description.</i>		
Page number:	17	
Number of Lines:	8	
Data description:	The King's Page 17. Create a Document, a Page Form or a Line Form from predefined Page-, Line- or Bit-Form Lists.	
<i>Line description.</i>		
Line 0:	City address	
Line 1:	00010001 (Page 17)	
Line 2:	r r r L L r r E E	List Extension: 0 Receive 1 Transmit
	LL	Target List type: 01 Line Form List 10 Page Form List 11 Document List
	r = 0	reserved bits
Line 3:	xxxxxxxx	Source List number
Line 4:	xxxxxxxx	Source record number
Line 5:	xxxxxxxx	Target List number
Line 6:	xxxxxxxx	Target record number
Line 7:	xxxxxxxx	Position on Target Form Number in a Document or Line Number on a Page Form or Bit Number on a Line.

8.1.14 The **King's Page 18**. Create a Compressed Form from Presentation Lists.

With the King's Page 18 the Kingdom Founder can create a Compressed Form from defined Presentation Lists (sources).

Form for the King's Page 18:

Document name:	King's Document	
Document List:	T1 Capital / 0 City	
Document Number:	0 Capital / 0 City	
Document type:	Transmit (Capital) Receive (City)	
<i>Page description.</i>		
Page number:	18	
Number of Lines:	8	
Data description:	The King's Page 18. Create a Compressed Form from predefined Lists.	
<i>Line description.</i>		
Line 0:	City address	
Line 1:	00010010 (Page 18)	
Line 2:	rrccrrL	L List Extension:
	0	Receive
	1	Transmit
	rr = 00	reserved bits
	ccc	Form type:
	000	Compressed Page Form
	001	Compressed Envelope Form, Std.
	010	Compressed Letter Form, Std
	101	Compressed Envelope Form, Ext.
	110	Compressed Letter Form, Ext.
	rr = 00	reserved bits
Line 3:	xxxxxxxx	Source List number
Line 4:	xxxxxxxx	Source record number
Line 5:	xxxxxxxx	Target Form List number
Line 6:	xxxxxxxx	Target record number
Line 7:	xxxxxxxx	Start Position (LSB) on target. (Signed Integer: -29 - -19, 0 - 63 (Std.) -29 - 63 (Ext.))

8.1.15 The **King's Page 19**. First Identifier mask

First Identifier masks are used for Compressed Envelopes described in paragraph 4.6.2, page 29. Examples on how to design such masks can be found in paragraph 4.7.1, page 32.

Form for the King's Page 19:

Document name:	King's Document	
Document List:	T1 Capital / 0 City	
Document Number:	0 Capital / 0 City	
Document type:	Transmit (Capital) Receive (City)	
<i>Page description.</i>		
Page number:	19	
Number of Lines:	8	
Data description:	The King's Page 19. Construction of a First Identifier mask.	
<i>Line description.</i>		
Line 0:	City or Group address	
Line 1:	00010011 (Page 19)	
Line 2:	xxxxxxx	Bit pattern, LSB
Line 3:	xxxxxxx	Bit pattern
Line 4:	xxxxxxx	Bit pattern
Line 5:	Errxxxx	Bit pattern, MSB
	r = 0	Reserved
	E = 1/0	Extended / Std. format.
Line 6:	A r r r r x x x	xxx Number of filtering primitive. An existing primitive will be overwritten.
	r = 0	Reserved
	A=1/0	Remove primitive yes/no.
Line 7:	xxxxxxx	Filter Number

8.1.16 The **King's Page 20**. Create filters for a Postmaster.

Construction of this type of filters is described in paragraph 4.7.2, page 37

Form for the King's Page 20:

Document name:	King's Document		
Document List:	T1 Capital / 0 City		
Document Number:	0 Capital / 0 City		
Document type:	Transmit (Capital) Receive (City)		
<i>Page description.</i>			
Page number:	20		
Number of Lines:	8		
Data description:	The King's Page 20. Filter construction.		
<i>Line description.</i>			
Line 0: City or Group address			
Line 1:	00010100	(Page 20)	
Line 2:	xxxxxxx	Bit pattern, LSB	(Bit -1 - -8)
Line 3:	xxxxxxx	Bit pattern	(Bit -9 - -16)
Line 4:	xxxxxxx	Bit pattern	(Bit -17 - -24)
Line 5:	Errxxxx	Bit pattern, MSB	(Bit -29 - -25)
		r = 0	reserved
		E = 1/0	Extended / Std. format.
Line 6:	r r r r r xxx	xxx	Filter primitive number. An existing primitive will be overwritten.
		r = 0	Reserved
Line 7:	xxxxxxx	Filter Number	

Recommended Page 20 settings for several different CAN controllers are found below (please see the respective data sheet for further details):

Intel 82526:

Not applicable.

Intel 82527:

Filter 6 (any primitive) : The bit pattern specified on Line 2 will be placed in register 06H. The bit pattern specified on Line 3 will be placed in register 07H.

(This is the Global Mask for Std. Messages)

Filter 8 (any primitive) : The bit patterns on Line 2 - 5 will be placed in register 08H - 0BH, respectively. (This is the Global Mask for Ext.. Messages)

Filter 12 (any primitive) : The bit patterns on Line 2 - 5 will be placed in register 0CH - 0FH, respectively. (This is the Message 15 Mask.)

Philips 82C200 :

Any filter, any primitive: The bit pattern on Line 2 will be placed in the Acceptance Code Register. The bit pattern on Line 3 will be placed in the Acceptance Mask Register.

NEC mPD72005:

Filter 0: The bit patterns on Line 2-5 will be placed in the MKA00-MKA03 registers, respectively.

Filter 1: The bit patterns on Line 2-5 will be placed in the MKA10-MKA13 registers, respectively.

Filter 2: The bit patterns on Line 2-3 (std CAN) or 2-5 (ext. CAN) will be placed in the identifier part of the buffer designated (by register GM0) as global buffer 0.

Std. CAN:

Line 2: x x x x ID₂₈ ID₂₇ ID₂₆

Line 3: ID₂₅ ID₂₄ ID₂₃ ID₂₂ ID₂₁ ID₂₀ ID₁₉ ID₁₈

Ext. CAN:

Line 2: ID₁ ID₀ x x x ID₂₈ ID₂₇ ID₂₆

Line 3: ID₂₅ ID₂₄ ID₂₃ ID₂₂ ID₂₁ ID₂₀ ID₁₉ ID₁₈

Line 4: ID₁₇ ID₁₆ ID₁₅ ID₁₄ ID₁₃ ID₁₂ ID₁₁ ID₁₀

Line 5: ID₉ ID₈ ID₇ ID₆ ID₅ ID₄ ID₃ ID₂

Filter 3: The bit patterns on Line 2-3 (std CAN) or 2-5 (ext. CAN) will be placed in the identifier part of the buffer designated (by register GM1) as global buffer 1. The Lines have the same layout as they have in Filter 2.

Motorola 68HC05X04,-X16:

Any filter, any primitive: The bit pattern on Line 2 will be placed in the Acceptance Code Register (CACC). The bit pattern on Line 3 will be placed in the Acceptance Mask Register (CACM).

For CAN Controllers not specified above, please contact KVASER.

8.2 The Mayor's Document.

Any City, including the Capital, has a “Mayor’s Document.” The first Page is optional and contains the Identification Number of the City (see page 99). The City Founder must prepare a Mayor's Document that identifies the City but may also have Pages reflecting current settings of different kinds. This Document belongs to Folder Number 1 in the Mayors office and contains at least two predefined Forms for Page 0 and 1 to ensure a safe identification of the City, currently the only ones specified in CAN Kingdom. Forms for Pages up to 127 are reserved but Pages 128 and above can be used by the City Founder if his Mayor needs to handle information for diagnostics, setups, etc. .

8.2.1 The Mayor's Pages 0 and 1 for City Identification.

In CAN Kingdom the two first Forms of the **Mayor's Document, Form for Page 0 and 1**, is used for **City Identification**. Page 0 contains the EAN-13 Code and Page 1 the serial number of the City. If more information is needed to identify a City, then City specific Pages above Page number 127 can be used for, e.g., identifying enabled software variant, an ASCII string, modules in a modular part number, etc.

Form for the Mayor's Page 0:

Document name:	Mayor's Document.		
Document List:	T1		
Document Number:	1.1		
Document type:	Transmit		
<i>Page description.</i>			
Page number:	0		
Number of Lines:	8		
Data description:	City Identification, EAN-13 Code.		
<i>Line description.</i>			
Line 0:	r r r r r r r r	r = 0	reserved
Line 1:	00000000		Page 0
Line 2:	xxxxxxx	LSB	
Line 3:	xxxxxxx		Product Identification Code
Line 4:	xxxxxxx		(EAN-13, Check-code omitted)
Line 5:	xxxxxxx		Unsigned 40-bit integer.
Line 6:	xxxxxxx	MSB	
Line 7:	r r r r r r r r	r = 0	reserved

Form for the Mayor's Page 1:

Document name:	Mayor's Document.	
Document List:	T0	
Document Number:	1.1	
Document type:	Transmit	
<i>Page description.</i>		
Page number:	1	
Number of Lines:	8	
Data description:	City Identification, Serial Number.	
<i>Line description.</i>		
Line 0:	r r r r r r r r	r = 0 reserved
Line 1:	00000001	Page 1
Line 2:	xxxxxxxx	LSB
Line 3:	xxxxxxxx	
Line 4:	xxxxxxxx	Serial Number
Line 5:	xxxxxxxx	Unsigned 40-bit integer.
Line 6:	xxxxxxxx	MSB
Line 7:	r r r r r r r r	r = 0 reserved

Form for the Mayor's Pages >127:

Document name:	Mayor's Document.		
Document List:	T1		
Document Number:	1		
Document type:	Transmit		
<i>Page description.</i>			
Page number:	n		
Number of Lines:	8		
Data description:	Mayor's Page, according to City specification.		
<i>Line description.</i>			
Line 0:	rrrrrrrr	r = 0	reserved
Line 1:	nnnnnnnn	Page n	127 < n > 254
Line 2:	xxxxxxx	Lines 2 - 7 according to City definition.	
Line 3:	xxxxxxx		
Line 4:	xxxxxxx		
Line 5:	xxxxxxx		
Line 6:	xxxxxxx		
Line 7:	xxxxxxx		

8.2.2 The Envelope assigned to the Folder 1, the Mayor's Document Folder.

At the start up of the Kingdom each Mayor must, as soon as possible, assign an Envelope to the Mayor's Document Folder:

This Envelope Number is the **Base Number+n**.

The **Base Number** is given by the Kingdom Founder and can be *transmitted to the Mayors from the King with the King's Letter, Page 0*, or provided by hardware settings¹. The number **n** is the physical number of the City in the Postal System.

¹ The physical number of a City (n) as well as the Base Number can be provided by a KVASER T-CANnector when the City is connected to the Postal System ref.[4].

8.2.3 The Mayor's Document Folder Label.

In a City the "Mayors Document Folder Label" must contain the following information:

1. Folder Number: 1.
2. Document List Number: T0
3. Document Number: 1
4. Transmit: Yes.
5. The CAN Control Field: 001000
6. Remote Envelope: Yes. (Only Page 0 is sent on a remote request.)
7. Enable the use of the Document in this Folder: Yes.
8. Envelope assigned to this Folder: Base Number + n.
9. Envelope enable: Yes.

To receive and decode these Letters from the Mayors, the King needs at least one Folder with a Mayor's Document. Assigned to this Folder are all the Envelopes used by the Mayors for this purpose.

8.3 Block Transfer Document.

The Forms for a Block Transfer Document are specified below. You find a description of Block Transfers in paragraph 7, page 55.

Form for the Block Transfer Page 0, Request Page:

Document name:	Block Transmit or Receive Transfer Document	
Document List:	Tx / Rx	
Document Number:	x	
Document type:	Transmit (Block receiving City) / Receive (Block transmitting City)	
<i>Page description.</i>	Request Page. (Only sent by the Block receiver.) The receiving Mayor requests the transmission of a Block.	
Page number:	0	
Number of Lines:	8	
Data description:	Only reserved data.	
<i>Line description.</i>		
Line 0:	rrrrrrrr	r = 0 reserved
Line 1:	00000000	(Page 0)
Line 2:	rrrrrrrr	r = 0 reserved
Line 3:	rrrrrrrr	
Line 4:	rrrrrrrr	
Line 5:	rrrrrrrr	
Line 6:	rrrrrrrr	
Line 7:	rrrrrrrr	

Form for the Block Transfer Page 1, Setup Page:

Document name:	Block Transfer Document	
Document List:	Tx / Rx	
Document Number:	x	
Document type:	Transmit (Block Transmitting City) / Receive (Block Receiving City)	
<i>Page description.</i>	Setup Page. (Only sent by the Block transmitter.) The transmitter informs the receiving Mayor about the length of the Block to be transmitted.	
Page number:	1	
Number of Lines:	8	
Data description:		
<i>Line description.</i>		
Line 0:	rrrrrrrr	r = 0 Reserved
Line 1:	00000001	(Page 1)
Line 2:	xxxxxxx	LSB
Line 3:	xxxxxxx	
Line 4:	xxxxxxx	
Line 5:	xxxxxxx	MSB Length of the Block in Lines (eight bit bytes). all x = 1 The length is indefinite.
Line 6:	rrrrrrrr	r=0 Reserved.
Line 7:	rrrrrrrr	

Form for the Block Transfer Page 2, Bundle Request Page:

Document name:	Block Transmit or Receive Transfer Document		
Document List:	Tx / Rx		
Document Number:	x		
Document type:	Transmit (Block Receiving City) / Receive (Block Transmitting City)		
<i>Page description.</i>	<p>Bundle Request Page. (Only sent by the receiver.) The receiving Mayor informs the transmitter about how many consecutive Pages he wants to receive before he transmits a new Bundle Request Page.</p>		
Page number:	2		
Number of Lines:	8		
Data description:	Number of Pages in a bundle.		
<i>Line description.</i>			
Line 0:	rrrrrrrr	r = 0	reserved
Line 1:	00000010		(Page 2)
Line 2:	nnnnnnnn	LSB	
Line 3:	nnnnnnnn	MSB	Number of Pages in the next Bundle. The last Page in the last Bundle must contain at least one byte of the Block.
		all n = 1	Send the complete Block in one Bundle.
		all n = 0	Session completed successfully.
Line 4:	rrrrrrrr	r = 0	Reserved.
Line 5:	rrrrrrrr		
Line 6:	rrrrrrrr		
Line 7:	rrrrrrrr		

Form for the Block Transfer Page 3, Information Transport Page:

Document name:	Block Transmit or Receive Transfer Document	
Document List:	Tx / Rx	
Document Number:	x	
Document type:	Transmit (Block Transmitting City) / Receive (Block Receiving City)	
<i>Page description.</i>	Information transport Page. A Block is divided into six byte packages and each package is transmitted by a Letter containing Page 3. Only sent by the transmitter.	
Page number:	3	
Number of Lines:	8	
Data description:	Six bytes of data to be transferred.	
<i>Line description.</i>		
The data type of Line 0 - 1 is Binary . Other Lines according to City specification.		
Line 0:	r r r r r r r r	r = 0 reserved
Line 1:	00000011	(Page 3)
Line 2:	xxxxxxxx	First data byte
Line 3:	xxxxxxxx	Second data byte.
Line 4:	xxxxxxxx	Third data byte.
Line 5:	xxxxxxxx	Fourth data byte.
Line 6:	xxxxxxxx	Fifth data byte.
Line 7:	xxxxxxxx	Sixth data byte.

The whole Block is stored in a set of consecutive Pages, alternately numbered 3 and 4. These Pages have no sub-pagination. If the last Line(s) is not filled with valid data, then they must be padded with zeros. If the latest Bundle is not accepted by the receiver, the transmission is aborted by an Abort Page, Page 5, from the receiver.

Form for the Block Transfer Page 4, Information Transport Page:

Document name:	Block Transmit or Receive Transfer Document	
Document List:	Tx / Rx	
Document Number:	x	
Document type:	Transmit (Block Transmitting City) / Receive (Block Receiving City)	
<i>Page description.</i>	Information transport Page. A Block is divided into six byte packages and each package is transmitted by a Letter containing Page 4. Only sent by the transmitter.	
Page number:	4	
Number of Lines:	8	
Data description:	Six bytes of data to be transferred.	
<i>Line description.</i>	The data type of Line 0 - 1 is Binary . Other Lines according to City specification.	
Line 0:	rrrrrrrr	r = 0 Reserved
Line 1:	00000100	(Page 4)
Line 2:	xxxxxxx	First data byte
Line 3:	xxxxxxx	Second data byte.
Line 4:	xxxxxxx	Third data byte.
Line 5:	xxxxxxx	Fourth data byte.
Line 6:	xxxxxxx	Fifth data byte.
Line 7:	xxxxxxx	Sixth data byte.

The whole Block is stored in a set of consecutive Pages, alternately numbered 3 and 4. These Pages have no sub-pagination. If the last Line(s) is not filled with valid data, then they must be padded with zeros. If the latest Bundle is not accepted by the receiver, the transmission is aborted by an Abort Page, Page 5, from the receiver.

Form for the Block Transfer Page 5, Abort Page:

Document name:	Block Transmit or Receive Transfer Document	
Document List:	Tx / Rx	
Document Number:	x	
Document type:	Transmit (Transmitting and Receiving City) / Receive (Transmitting and Receiving City)	
<i>Page description.</i>	Abort Page. (Sent by transmitter or receiver.) The transfer is aborted.	
Page number:	5	
Number of Lines:	8	
Data description:		
<i>Line description.</i>		
Line 0:	rrrrrrrr	r = 0 reserved
Line 1:	00000101	(Page 5)
Line 2:	rrrrrrrr	r = 0 reserved
Line 3:	rrrrrrrr	
Line 4:	rrrrrrrr	
Line 5:	rrrrrrrr	
Line 6:	rrrrrrrr	
Line 7:	rrrrrrrr	

8.4 Time Herald Document.

This one Page Document is used for synchronizing the clocks at each City in the Kingdom against one or more master clocks. The Time Herald may be the King or a specific Mayor but other schemes for synchronization, using more than one Time Herald, might offer a better security.

Form for the Time Herald Page:

Document name:	Time Herald Document	
Document List:	x	
Document Number:	x	
Document type:	Transmit (Time herald) Receive (City)	
<i>Page description.</i>		
Number of Lines:	4	
Data description:	Circular Time message.	
<i>Line description.</i>		
Line 0:	00000000	Time message.
	00000001	Prepare for time message.
	00000010	Actual source time at Time message.
Line 1:	gggggggg	Grain number.
Line 2:	ssssssss	Segment number.
Line 3:	rrrrrrrr	Revolution number.

8.5 Linear Time Document.

Form for the Linear Time Page:

Document name:	Linear Time Document	
Document List:	x	
Document Number:	x	
Document type:	Transmit (Time herald) Receive (City)	
<i>Page description.</i>		
Number of Lines:	8	
Data description:	Linear Time message.	
<i>Line description.</i>		
Line 0:	mmmmmmmm	Millisecond, LSB.
Line 1:	mmmmmmmm	Millisecond, MSB. (0 - 999) All m = 1 -> Do not care.
Line 2:	s s s s s s s s	Second. (0 - 59) All s = 1 -> Do not care.
Line 3:	MMMMMMMM	Minute. (0 - 59) All M = 1 -> Do not care.
Line 4:	HHHHHHHH	Hour. (0 - 23) All H = 1 -> Do not care.
Line 5:	DDDDDDDD	Day, LSB.
Line 6:	DDDDDDDD	Day, MSB. (0 - 365) All D = 1 -> Do not care.
Line 7:	YYYYYYYY	Year. (0 - 99) All Y = 1 -> Do not care.

9. Start-up procedure.

For a safe function of the Postal System in a Kingdom it is essential that any City acts in a sound and safe way when connected to the Postal System. A City can destroy the communication in two major ways:

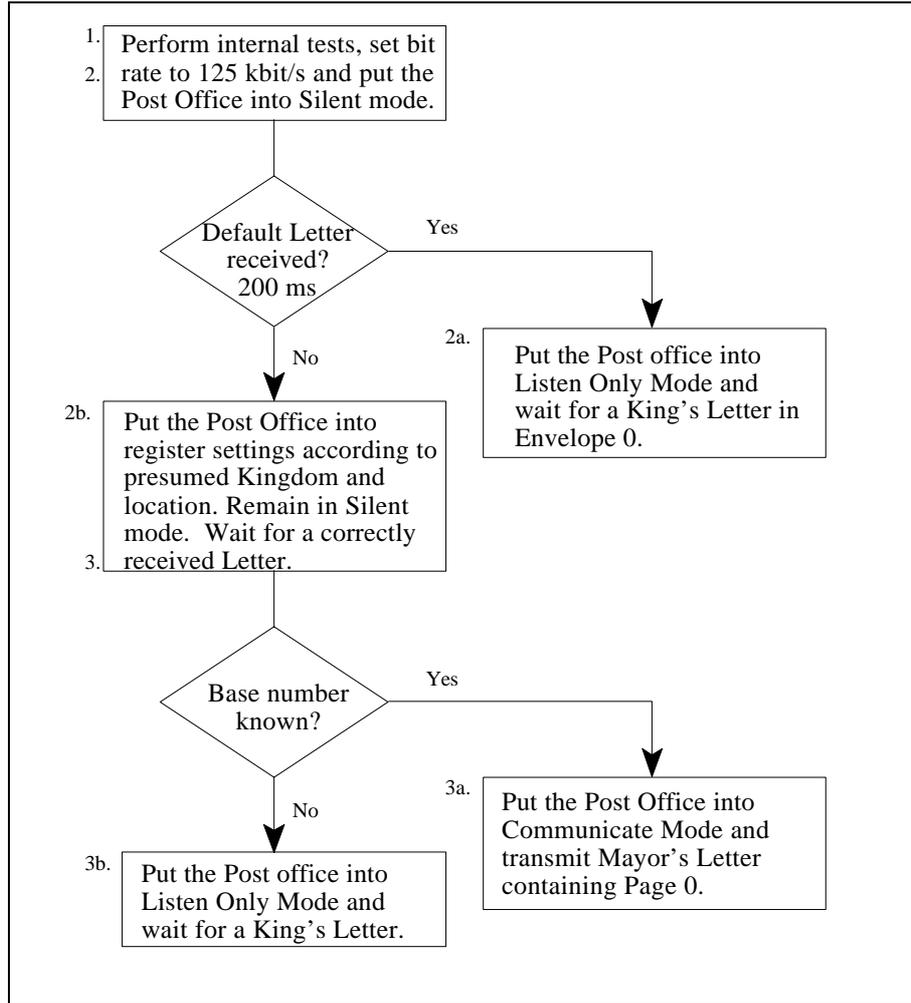
1. By using another baud rate than the Postal System.
2. By transmitting a Letter in an Envelope assigned for transmission of another City.

To avoid destroying the communication by accident, a City will keep silent until its Postmaster has received a correct Letter. If the Mayor knows the Base Number of the Kingdom he announces his presence, otherwise he will wait to be polled by the King.

As the bit-timing register settings can be downloaded by the King, the register settings might be accidentally corrupted during the download procedure. If this happens it can be impossible to get connected again. To assure a possibility to always get connected, any City must use a default register setting during the first 200 milliseconds at start-up.

The following start-up procedure is prescribed for CAN Kingdom Cities:

1. The Mayor performs internal tests.
2. The Mayor puts the Post Office into a bit rate of 125 kbit/s and Silent Mode (no acknowledgement bits nor error frames) for 200 ms waiting for a Default Letter with Envelope 2031_{Std} containing a Page with all eight Lines containing AA_{hex}.
 - 2a If such a Letter is received during the 200 ms period, the Mayor keeps the time register settings and switches to Listen Only Mode waiting for a King's Letter **in Envelope 0**.
 - 2b If no Default Letter is received during the 200ms period, the Mayor puts the Post Office into register settings according to presumed Kingdom and location, still in Silent Mode, waiting for a correctly received Letter.
3. When a Letter is correctly received:
 - 3a If the Base Number is known, the Mayor change the Post Office into Communication Mode and transmits his Mayor's Letter with Page 0.
 - 3b If the Base Number is **not** known, the Mayor switches to Listen Only Mode and waits for a King's Letter.



10. CAN Kingdom Product Identification Rules.

The identification of a City is basically founded on EAN 13-codes and a numeric 12 digit serial number.

The EAN system is unambiguously transparent to the UPC (Uniform Product Code) system used in the US and Canada. The Product Identification Code (PID) is a producer unique code consisting of a 40 bit unsigned integer based on the EAN-13 identification code. Thus, the decimal value of PID is less than

1,099,511,627,776.

The first to the fourth digit (*italic*) is used for the EAN numeric-3 digit prefix P1, P2, P3 which identifies the Numbering Organization. The next four digits (**bold**) are a company identification issued by the EAN Numbering Organization. The remaining five digits are a company specific product identification code. The total code is then identical with the EAN-13 code of a product but the Check Digit omitted.

Identification Number.

The Identification Number of a City is the EAN-13 code but the check digit. In the Mayor's Document, Page 0, the Identification Number is written as an unsigned 40-bit integer.

Serial Number.

Although the EAN rules accept alphanumeric serial numbers, **only numeric serial numbers are used in CAN Kingdom** and have to be less than 1,099,511,627,776. In the Mayor's Document, Page 1, the Serial Number is written as a 40-bit unsigned integer.

Further identification.

If further identification, e.g., an ASCII string, is required, then such codes can be placed in City specific Mayor's Pages according to Forms above 127.

11. Graphical presentation of a CAN Kingdom.

On the following pages an example of a simplified CAN Kingdom is presented in a graphical form. The Kingdom has one Capital and four Cities:

City 1.

The City 1 is a controller City and needs temperature values, presented in a floating point format, from two sensor Cities to perform its control task. The City is designed to be placed into different Kingdoms and will behave differently in various Kingdoms.

City 2.

The City 2 is a display unit that can present two different values. The presented values will be received as integers.

City 3 and 4.

These Cities are identical and each measures one temperature. The temperature measured can be presented as a floating point value or as an integer.

Figure 9 describes the setup procedure. All Cities receive King's Letter in Envelope 0. When they receive King's Page 1 they find the Base Number, in this case 40, and respond then with their Mayor's Letter in Envelopes 41, 42, 43 and 44 respectively. The City 1 that needs information about the Kingdom it currently belongs to, receives a Mayor's Letter from the Capital in Envelope 40.

A comment on the Capital:

The Kingdom Founder has chosen to instruct the King to assign the Envelopes for the different Mayor's Letters to individual Folders, each containing a copy of the Mayor's Document. Another solution might have been to receive all Mayor's Letters in one single Folder. An advantage with the chosen solution is that when the King receives a Mayor's Letter he immediately knows from which position in the Kingdom it originates. If he only has one Folder, he has to check the Envelope number of the received Letter in order to get the position number of the originator.

Now everything is ready for the King to distribute Envelopes by King's Page 2. **Figure 10** shows the result of this procedure. City 1 receives Temperature 1 from City 4 as a floating point value in Envelope 80 and in the same way Temperature 2 from City 3 in Envelope 81. The display unit (City 2) receives the temperature values as integers in Envelope 90 and 91. After sending King's Page 0, run, the Capital and the King are not needed anymore.

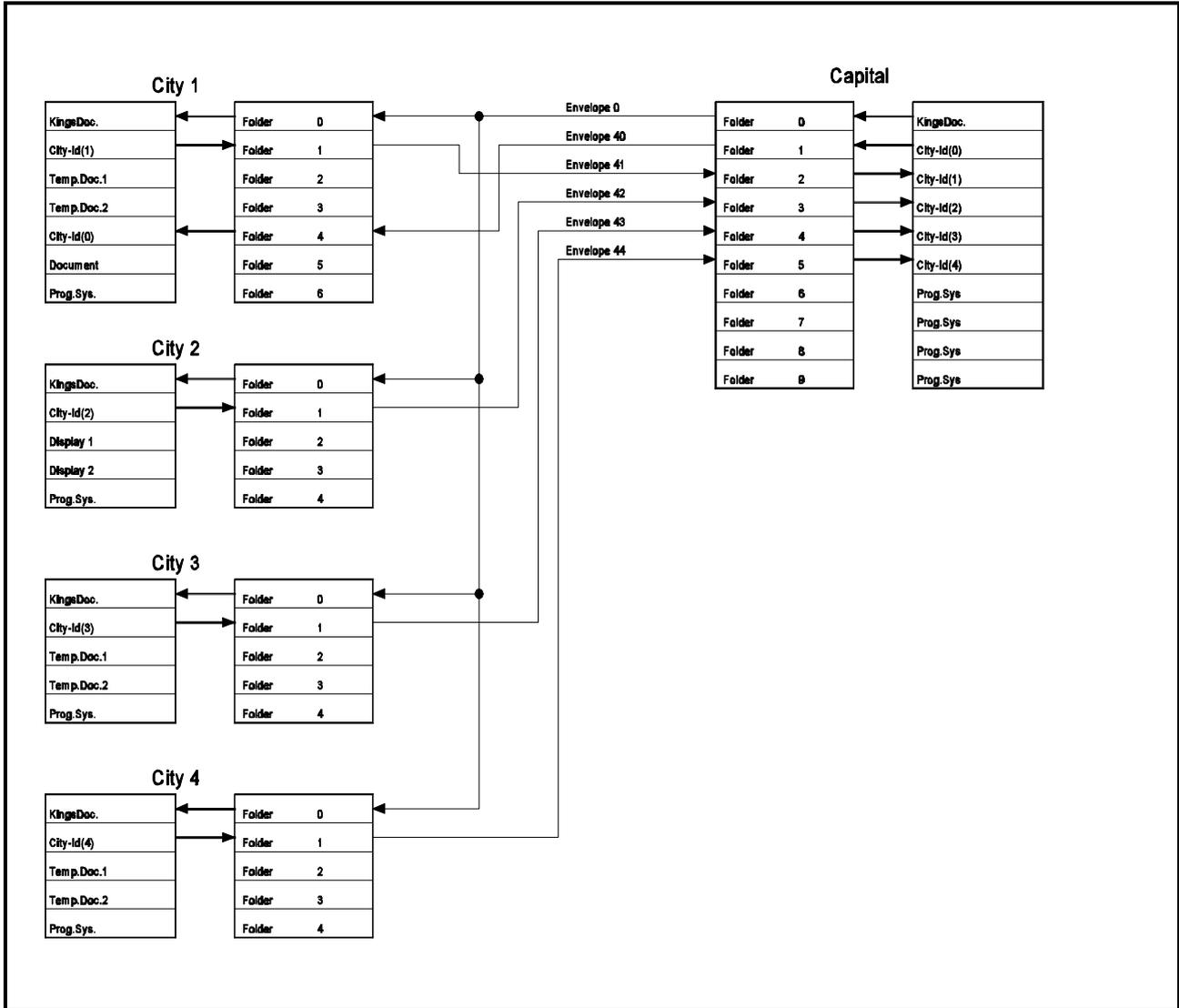


Figure 9. A CAN Kingdom organized for the set-up phase.

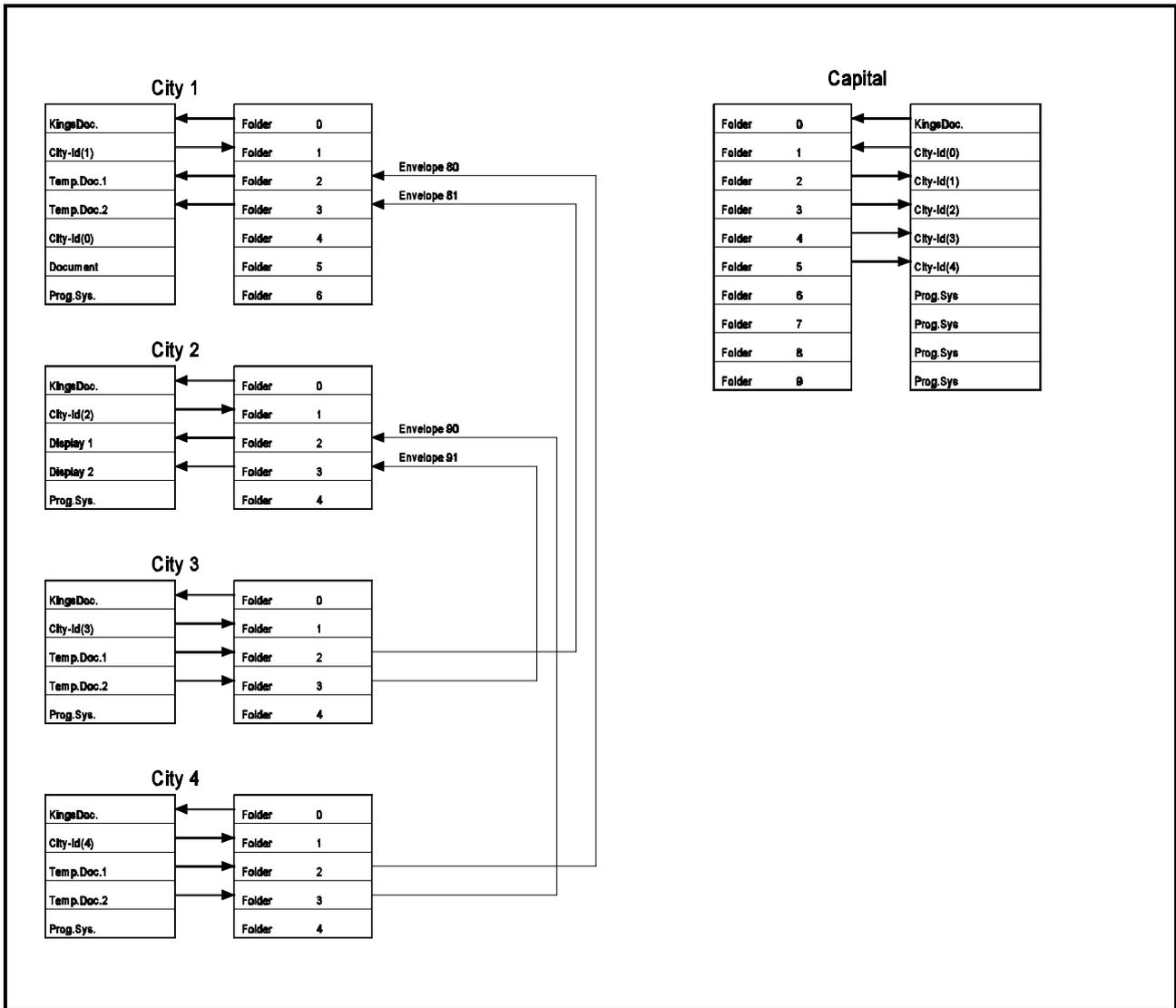


Figure 10. A CAN Kingdom organized for run time phase.

12. CAN Kingdom Glossary.

CAN Simile words are capitalized.

CAN Simile expression	Corresponding CAN expression or function
Bit	Bit in a byte in the CAN data or identifier field.
CAN Kingdom Founder	System designer.
Capital	Supervisor or network managing module in a system.
City	Complete CAN module, e.g., an actuator.
City Founder	Module designer.
City Identification Letter	CAN object used to identify modules connected to the network.
Document	A written single Page or numbered written Pages that will be sent one by one in the same Envelope.
Envelope	CAN identifier.
Envelope 0	CAN Identifier for Letter 0, reserved for the supervisor module. Called the King's Envelope.
Folder	Link between a Document and the Postal System.
Folder Label	Information needed to link a Document to the Postal System.
Form	Description of the Lines on a Page.
Group	A cluster of Cities that have a common address.
King	μ C program in the supervisor module.
Kingdom	A complete system with a CAN network with a supervisor module.
Kingdom Founder	System designer.
King's Letter	The Letter with Envelope 0
Letter	A complete CAN message.
Line	Data byte in a CAN data or identifier field.
List	A table of Variable Envelopes, Documents, Forms, Lines or Bits to be received or transmitted.
Mayor	μ c program in a CAN module.
Nation	A complete system with actuators, sensors, etc., connected by a CAN network.
Null Document	A Document with no Lines.

CAN Simile expression	Corresponding CAN expression or function
Page	CAN data field.
Paginated Page	CAN data field where one byte is the Page number.
Postal System	The CAN network.
Postal Route	The CAN bus signal path.
Postbox	DPRAM area for communication between the Communication Controller and the μ c on a CAN module.
Postman	Symbol for bit-wise arbitration.
Postmaster	Symbol for CAN Controller services.
Post Office	Communication Controller chip.
Round	A CAN transmission cycle.

13. References.

1. PCT 85/01007 "Arrangement Comprising a System Providing Movement, Processing and/or Production".
Date: 14 March 1985.
2. Mark S. Fox, "An Organizational View of Distributed Systems." IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-11, No. 1, January 1981.
3. Neal Laurance, "Connecting CAN-based Networks to an MMS Network", CiA ICC'94 proceedings, page 4-4, CAN in Automation 1994.
4. PCT WO/ 92/03881 "Distributed Computer System Arrangement".
Date: 5 March 1992.
5. ISO 11898 "Road vehicles - Interchange of digital information - Controller area network (CAN) for high-speed communication"
6. Fredriksson, L-B ; Serielles Steuergeräte-Netzwerksystemfür Webmaschinen - der Schlüssel zur Vollautomation; 6. Weberei Kolloquium Denkendorf 1990, ITV Institut für Textil- und Verfahrenstechnik Denkendorf, Germany.
7. PCT WO 91/20019 "Device for Controlling a Member in a System".
Date: 26 December 1991.
8. PCT WO 90/13078 "Arrangement in a Computer System".
Date: 1 November 1990.
9. Ken Tindell, Alan Burns, "Guaranteed Message Latencies For Distributed Safety-Critical Hard Real-Time Control Networks." Technical Report YCS229, University of York, Dept. Computer Science, 1994.
10. PCT WO 91/10960 "Arrangement for a Distributed Control System".
Date: 25 July 1991.

Appendix

A note on Compressed Forms.

As the reader probably knows, there are two different ways of storing multi-byte pieces of information in memory: “little endian” and “big endian,” the former being supported by Intel (on certain platforms), Digital, and others and the latter being supported by Motorola et al.

Proponents of the “little endian” approach would store, say, a 4-byte quantity in the memory starting with the least significant byte on the lowest memory address, continuing with more significant bytes on higher memory addresses.

Those who advocate the “big endian” way of doing things would store the same quantity starting with the most significant byte on the lowest address, proceeding with the less significant bytes on higher memory addresses.

This gives us a headache when we try to store quantities across byte boundaries, for example when trying to pack a Letter as full as possible. CAN Kingdom, being slightly in favor of the little-endian camp, defines the “Start Position” of a quantity on a Form as the bit number of the Least Significant Bit. Further, CAN Kingdom numbers the bits in the following way:

	Most significant bit within a byte ----- least significant bit within a byte							
Byte 0 (least significant)	7	6	5	4	3	2	1	0
Byte 1	15	14	13	12	11	10	9	8
Byte 2	23							16
Byte 3	31							24
Byte 4	39							32
Byte 5	47							40
Byte 6	55							48
Byte 7 (most significant)	63							56

Assume that we want to store a 4-bit and an 8-bit quantity in a compressed Form. If we have an all-Intel environment, we would like to favorize little-endian and so ease the encoding and decoding in the source and target Cities. A “C” programmer would like to define just a struct and then just read or write the values directly to a field in this struct. Consequently, we write

```

struct    foo {
    int a: 4;
    unsigned b: 8;
} bar;

```

and then we can write

```

bar.a = 6;
bar.b = 42;

```

to get the desired result. Now, in the Intel environment, the struct foo will occupy the following bits (the diagonal lines mark the a field and the gray fields mark the b field):

Bit7			LSB	MSB			
Bit15				MSB			bit8
Bit23							bit16
Bit31							bit24

but in an Motorola environment, the following bits will be used:

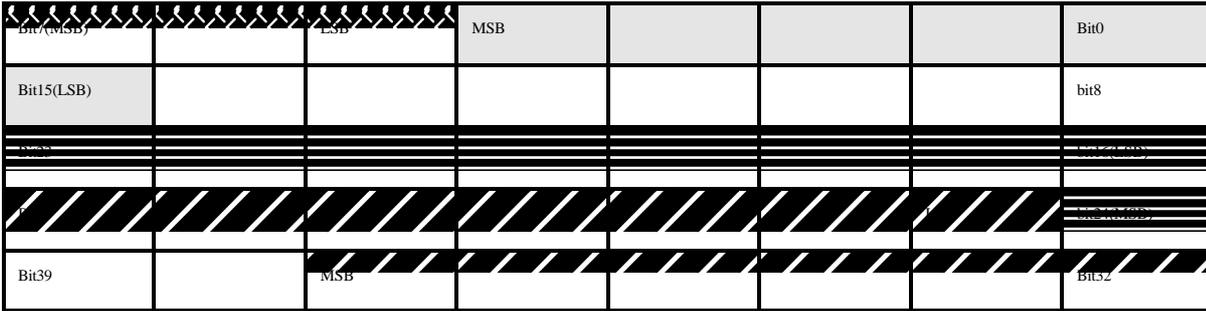
MSB							Bit0
Bit15			LSB				bit8
Bit23							bit16
Bit31							bit24

Here, we can see two problems:

- The Start Position varies with the endian-ness of the variables. In the Intel example, the start position of a and b is 0 and 4, respectively. In the Motorola example, however, the start positions are 4 and 12! Obviously, we have to know the endianness of a quantity *even if it does not occupy more than a full byte*.
- Sending data between two Cities with different native endianness can be a bit tedious - somebody has to carry the burden of converting the quantity to its native format.

While we are at the subject, please note that compilers usually impose constraints on the usage of bit fields. Most will not accept a bit field bigger than some number of bytes, usually 2 or 4.

Another example: assume we want to pack a) a 3-bit variable (big endian), b) a 9-bit variable (little endian), c) a 13-bit variable (little endian), and d) a 6-bit variable (big endian) as tightly as possible into a Form. The layout of the bits will be this:



The King specifies the Start Positions to 5, 15, 16, and 25 (i.e. the positions of the least significant bits.)
 The City knows the endianness of the variables, so their positions on the Form is uniquely determined.

The “C” programmer can now use the following struct on a big-endian platform:

```
struct {
    int a:3;
    int b:6;
    int :7;
    unsigned d1:8;
    unsigned c1:6;
    unsigned d2:1;
    unsigned c2:6;
} foo;
```

and conveniently access the a and b fields just by saying “foo.a” and “foo.b”, but to use the two other fields he must write “foo.c1 + (foo.c2<<6)” and “foo.d1 + (foo.d2<<8)” respectively.

In the little-endian City, the following struct is used:

```
struct {
    unsigned b1:5;
    unsigned a:3;
    int :7;
    unsigned a2:1;
    unsigned d:9;
    unsigned c:13;
} bar;
```

The d and c variables are easy to read and as a does not straddle a byte boundary we can access it directly too.

The hole in the middle of the Form appears since we try to pack both big-endian and little-endian quantities on the same Form. If we use only one kind, we can fill the Form completely at the expense of (possibly) increasing encoding or decoding complexity.

A final note: compressed Forms, especially those containing variables with mixed endian types, are highly specialized tools for use by advanced Cities only.