



G00001

IDENTIFICATION SYSTEM

THIS INVENTION relates to electronic tags. It also relates to a method of communicating data from an electronic tag, to an identification system, and to a receiver for receiving a transmission from the electronic tag.

According to the invention, there is provided an electronic tag which includes

processor means programmed to provide a modulation control signal which includes unique identification data which at least identifies the tag; and

transmitter circuitry connected to the processor means and to an antenna for transmission of the unique identification data, the transmitter circuitry being powered by the modulation control signal.

Preferably, the transmitter circuitry is exclusively powered by the modulation control signal of the processor means. Accordingly, the transmitter circuitry is not connected to another source of power but merely to ground and to the processor means. Thus, when the modulation signal is at 0 volts, the transmitter circuitry receives no power resulting in no transmission at all thereby enhancing the low power consumption characteristics of the electronic tag. There is thus no carrier wave and the output or identification signal is thus a pulsed wave switched between 0 volts and its maximum amplitude.



2

The transmitter circuitry may include passive components and a transistor defining oscillation circuitry directly driven by the processor means. The transistor in combination with the passive components may form an integral part of the transmitter circuitry which is powered up by the modulation signal.

The processor means may be configured to provide the modulation control signal with a first part followed by a second part. The first part may include at least one high pulse of such a duration so as to provide sufficient power to the transmitter circuitry at least partially to stabilize it for transmission of the second part. The second part may include data defined in a plurality of pulses which are of a substantially lesser duration. The signal transmitted by the transmitter circuitry of the tag resembles a combination of an amplitude modulated signal and a pulse width modulated signal. The amplitude modulation of the transmitter under control of the processor means is typically between about 0 % and about 100%. Accordingly, the power consumed by the transmitter whilst data is not being transmitted is substantially reduced. It is however to be appreciated that the transmitter circuitry may modulate the amplitude of the identification signal at any percentage between 0 % and 100 % thereby representing a plurality of values or levels and not merely two levels of "1"s and "0"s.

The modulation control signal may include a plurality of high pulses that, in combination, provide an identification signal to a tag receiver for receiving a transmission from the electronic tag. The high pulses are typically about 60 microseconds in duration with a 50 % duty cycle. It is however to be appreciated that the duty cycle and/or the duration may differ in various embodiments of the invention.

Thus, the high part of the modulation signal may perform a dual function. Firstly, it may power up the transmitter circuitry between, preferably, a totally switched off or dormant state, to an operative state in which it has stabilized sufficiently to transmit the second part comprising a pulse train of pulses of a substantially shorter duration. Secondly, the first part allows the receiver to distinguish a transmission from the tag from any other transmission e.g. an interference signal or the like from another source.

Each pulse of the second part of the modulation signal may include a start portion for identifying a start of a bit and a data portion for identifying a state of the bit of data. The duration of the data portion may selectively define a high and a low state of the bit under control of the processor means. The pulse width may be defined as the sum of the start and data portions.

The high state of the bit is typically defined by a shorter data portion during which the oscillator circuitry is switched off and the low state of the bit is defined by a longer data portion during which the oscillator circuitry is switched off.

The shorter pulse may be used to mark or identify the start of a bit after which the transmitter circuitry is switched off totally. The time interval or duration until the transmitter circuitry is switched on again defines the high or low state of the bit. The amount of power required to transmit a high bit and a low bit is substantially the same since power is only consumed to identify the start of a bit of data.

The processor means is typically a micro-controller which includes an internal RC oscillator on which the modulation control signal is dependent and the micro-controller is arranged to enter a sleep mode between data transmissions thereby to reduce power consumption.

5 Accordingly, the transmitter circuitry, under control of the processor means may be arranged periodically to transmit the identification signal is bursts at a predetermined time interval, for example about 1s. Preferably, the identification signal has a duty cycle of about 50 %.

10 Data is typically transmitted in a digital fashion as a series of "1"s and "0"s. Typically, a "0" is transmitted by a transmitter on time being about 10 microseconds followed by an off time of equal duration, and a "1" is transmitted by the transmitter being on about 5 microseconds followed by an off time of equal duration so that the signal has a 50 % duty cycle. It is however to be appreciated that any two
15 different transmission time intervals, controlled by the transmitter processor may be used to communicate a "1" or a "0". Further, the duty cycle of the pulses may vary.

Further in accordance with the invention, there is provided an identification system which includes

20 a plurality of electronic tags, each tag including
processor means programmed to provide a modulation control signal which includes unique identification data which at least identifies the tag; and
transmitter circuitry connected to the processor means and
25 to an antenna for transmission of the unique identification data, the transmitter circuitry being substantially powered by the modulation control signal; and

at least one electronic tag receiver configured to receive a transmission from the tag.

5 The transmitter circuitry of the electronic tag may be exclusively powered by the modulation control signal of the processor means.

10 The transmitter circuitry may include passive components and a transistor directly driven by the processor means. The transistor in combination with the passive components may form an integral part of the transmitter circuitry which is powered up by the modulation control signal.

15 The processor means may be configured to provide the modulation control signal with a first part followed by a second part, the first part including at least one high pulse of such a duration so as to provide sufficient power to the transmitter circuitry at least partially to stabilize for transmission of the second part which includes data defined in a plurality of pulses which are of a substantially lesser duration.

20 The first part of the modulation control signal may include a plurality of high pulses that, in combination, provide an identification signal to signal detection means of the electronic tag receiver for receiving a transmission from the electronic tag.

Each pulse of the second part of the modulation signal may include a start portion for identifying a start of a bit and a data portion for identifying a state of the bit of data, the duration of the data portion

selectively defining a high and a low state of the bit under control of the processor means.

The high bit may be defined by a shorter data portion during which the transmitter circuitry is switched off and the low bit is defined
5 by a longer data portion during which the transmitter circuitry is switched off.

Still further in accordance with the invention, there is provided a method of communicating data from an electronic tag, the method including driving transmitter circuitry of the tag with a
10 modulation control signal which substantially powers the transmitter circuitry.

Typically, the modulation control signal exclusively powers the transmitter circuitry.

The transmitter circuitry may include an oscillator which is
15 arranged to oscillate at its fundamental frequency when data is being transmitted and stop oscillating when data is not transmitted. Accordingly, the method may include selectively modulating a fundamental frequency of the oscillator when data is being transmitted and disabling the oscillator when data is not being transmitted.

20 The modulation control signal may include a first part followed by a second part, the first part including at least one high pulse of such a duration so as to provide sufficient power to the transmitter circuitry at least partially to stabilize it for transmission of the second

part which includes data defined in a plurality of pulses which are of a substantially lesser duration.

5 The first part of the modulation control signal may include a plurality of high pulses that, in combination, provide an identification signal to a tag receiver for receiving a transmission from the electronic tag.

10 Each pulse of the second part of the modulation signal may include a start portion for identifying a start of a bit and a data portion for identifying a state of the bit of data, the duration of the data portion selectively defining a high and a low state of the bit under control of the processor means.

15 The high bit may be defined by a shorter data portion during which the modulation control signal is switched off and the low bit may be defined by a longer data portion during which the modulation control signal is switched off.

20 The processor means is preferably a micro-controller which includes an internal RC oscillator on which the modulation control signal is dependent and the micro-controller is arranged to enter a sleep mode between data transmissions thereby to reduce power consumption. The micro-controller may define a transmitter processor which is typically a PIC 12C509 or the like, which is programmed with appropriate software to execute the method of controlling the transmitter.

Still further in accordance with the invention, there is provided a receiver for receiving a transmission from one of a plurality of

electronic tags, the transmission including a first part and a second part and the receiver including

5 detection circuitry for detecting the first part and the second part of the transmission, the first part including at least one high pulse in response to which the receiver monitors reception of the second part which includes data defined in a plurality of pulses which are of a substantially lesser duration; and

10 timing means for timing the duration of each of the pulses in the second part and selectively generating a high or a low output defining a bit dependent upon the duration of the pulse.

The receiver may include pulse width detection circuitry for decoding the identification signal.

The receiver may include

15 receiver circuitry connected to an antenna for receiving the identification signal from at least one electronic tag;

 demodulation circuitry connected to the receiver circuitry for demodulating the identification signal;

 amplification circuitry connected to the demodulation circuitry via a capacitive link; and

20 receiver processor circuitry connected to the amplification circuitry for processing the identification signal after demodulation thereof.

25 The receiver may include a repeater transmitter for re-transmitting the identification signal to a central control unit. Typically, each tag is attached to an item of value, e.g. a personal computer or other valuable item, located in a particular zone and a receiver monitors the transmission of identification signals in the zone. The central control

unit may thus be in wireless communication with a plurality of zones each of which include a receiver monitoring associated tags located on valuable items or equipment in the zone.

5 The invention is now described, by way of example, with reference to the accompanying diagrammatic drawings.

In the drawings,

Figure 1 shows a schematic circuit diagram of a electronic tag in accordance with the invention;

10 Figure 2 shows a schematic circuit diagram of a receiver, also in accordance with the invention;

Figure 3 shows a flow chart of a method of controlling transmission of data via the tag of Figure 1;

Figure 4 shows an example of a burst of data transmitted by the transmitter; and

15 Figure 5 shows a flow chart of a method of decoding data by the receiver of Figure 2.

Referring to the drawings, an identification system, in accordance with the invention, is provided which includes a plurality of transmitters which are in the form of electronic tags 10 (also in accordance with the invention and only one of which is shown in the drawings), each of which is associated with at least one receiver 12 (see Figure 2). Typically, valuable items or equipment to be monitored in a selected zone, e.g. computers in a particular office area, are each fitted with a tag 10 and the receiver 12 is located in the zone to monitor signals received from the tag 10. The receiver 12 forms part of a network of receivers which may be installed in a particular building or the

20

25

10

like. Each receiver 12 communicates via its repeater transmitter 14 to a central control unit (not shown). The central control unit may thus monitor and record authorised and/or unauthorised removal of the equipment.

5 Each tag 10 includes transmitter processor means in the form of a micro-controller 16 with associated support circuitry 18, and a long-life lithium battery 20. Selected I/O ports of the micro-controller 16 are connected to a connection terminal 22 via which resident software to control the method of transmission of data is programmed
10 into the micro-controller 16. In use, a reed switch 24 is selectively enabled to toggle the micro-controller 16 into various states or modes of operation. In one mode of operation, the reed switch 24 may function as a sensing means, for example, to sense movement or the like. In another mode of operation, the reed switch 24 may function as data
15 input means for feeding data into on-board memory of the micro-controller 16, e.g. data to uniquely identify the tag 10. The micro-controller 16 controls operation of transmitter circuitry 26 which transmits data, sourced from the micro-controller 16, to the receiver 12.

 As can clearly be seen from the schematic circuit diagram
20 shown in Figure 1, the micro-controller has one of its output ports connected directly to a transistor 25 of the transmitter circuitry 26. The transmitter circuitry 26 also includes associated passive components 27 which, in combination, define oscillator circuitry. Further, the transmitter circuitry 26 is not connected to the battery 20 but merely to ground and
25 is powered exclusively by the modulation control signal from the micro-controller via line 28. Thus, the transmitter circuitry 26 is toggled between an "on" state in which it transmits a pulse via its antenna 29

and an "off" state in which it receives no power at all and is thus switched off completely.

In order to allow the transmitter circuitry 26 to stabilize for transmission of data, the micro-controller 16 feeds three pulses (part of one pulse 31 being shown in Figure 4), each pulse having a high portion of between about 20 and about 70 microseconds, typically about 60 microsecond, and a duty cycle of 50 %, to the transmitter circuitry via line 28. In particular, the modulation control signal generated by software of the micro-controller 16 has a first part followed by a second part. The first part includes the three high pulses of about 60 microsecond duration which are generated by the routine "pulse 4" in Table 2. As shown by arrow 150 in Table 2, the "pulse 4" subroutine (see arrow 152) is called three times. The high part of the modulation signal performs a dual function. Firstly, it powers up the transmitter circuitry 26 between, preferably, a totally switched off or dormant state, to an operative state in which it has stabilized sufficiently to transmit the second part comprising a pulse train of pulses of a substantially shorter duration. Secondly, the first part allows the receiver to distinguish a transmission from the tag from any other transmission e.g. an interference signal or the like from another source.

The micro-controller 16 controls transmission of unique identification data in the second part of the modulation control signal by means of a method, also in accordance with the invention, carried out in its resident software. The method used by the micro-controller 16 is set out in Table 2 (see arrow 154). In particular, the micro-controller 16 controls the transmitter circuitry 26 in such a fashion so that its transmitted or identification signal is both amplitude modulated and a

pulse width modulated. The amplitude modulation of the transmitted signal varies between about 0% and about 100% modulation when representing both "1" and "0" (see Figure 3). The transmitted signal has a 50% duty cycle and a "1" is transmitted, for example, by a pulse with a 5 microseconds "on" time followed by a 5 microsecond "off" time resulting in a total pulse width of 10 microseconds (see Figure 4). A "0", on the other hand, is transmitted by a pulse with an "on" time of 10 microseconds and an "off" time of 10 microseconds resulting in a total pulse width of 20 microseconds. The total duration of the pulse, i.e. either 10 microseconds or 20 microseconds, that determines the state of a bit (see arrow 156 and following in Table 2). A plurality of pulses are transmitted in a burst of data and the burst of data is typically transmitted by the tag 10 periodically at a time interval of about 1 second. It is however to be appreciated that the pulses of the second part need not necessarily have a 50 % duty cycle since the high state of the pulse 33 (see Figure 4) acts as a marker or start portion identifying the start of a bit. Thereafter a data portion defines the state of the bit, the duration 35 of the data portion being the total width of the pulse including its "off". In the present embodiment, the data portion 35 defines a high or low state of the bit by a 10 microsecond transmitter circuitry "off" time and a 20 microsecond transmitter circuitry "off" time respectively (see Figure 3). When the receiver 12 receives a transmission from the tag 10, it determines the total length or duration of the pulse and assigns a "1" or "0" as shown in Figure 5.

Typically, the micro-controller 16 includes a counter which has its count reset upon installation of the battery 20 and thereafter increments its count each time the transmitter circuitry 26 transmits a burst of data. Data from the micro-controller 16 is fed to the transmitter

circuitry 26 via line 28. It is to be appreciated that the data transmitted by the tag 10 may include a value of the count, unique identification data for identifying the tag 10, data sensed by the reed switch 24, or any other data.

5 The micro-controller 16 of the tag 10 does not use a crystal oscillator to control its operation but uses an internal RC oscillator provided in the chip. It is believed that the power consumption of the tag 10 is thereby reduced and start-up delays are faster. Accordingly, in order further to reduce the power consumption of the tag 10, the tag
10 10 is dormant or asleep between each burst of data which it transmits. In order to achieve this, a second sleep or stand-by RC oscillator provided in the micro-controller 16, is used. As shown in block 30, after the predetermined time interval, typically about 0.7 to about 1s (see arrow 158 in Table 2) has lapsed, the micro-controller 16 is instructed
15 to start/wake-up. Thereafter, the unique identification code and other data to be transmitted by the tag 10 is set up as shown in block 32, whereafter the data is configured in a serial form as shown in block 34 where the next/first byte of the remaining bytes (designated by X) is fed to the transmitter circuitry 26 as shown in block 36. The micro-
20 controller 16 then (see block 38) analyses each bit in the serial string. If the bit is at logic "1", the transmitter circuitry 26 is activated to transmit a pulse which is high for 5 microseconds as shown in block 40 and, thereafter, a low of 5 microseconds is implemented as shown in block 42 to ensure that the transmitted signal has a 50% duty cycle and
25 a total pulse width of 10 microseconds. However, if the bit is at logic "0", the transmitter circuitry 26 is then instructed to transmit a pulse which is high for 10 microseconds as shown in block 44 followed by a low of 10 microseconds, as shown in block 46, providing a 50% duty

14

cycle and a total pulse width of 20 microseconds when a "0" is transmitted.

If all the bits of the byte have been transmitted, as shown in block 48, then the micro-controller 16 is instructed to increment its pointer to the next byte as shown in block 50 (see also arrow 154 in Table 2). If, however, all 8 bits of the byte have not been transmitted, then the micro-controller 16 is instructed to fetch the next bit as shown in block 52 and the procedure as set out above is repeated. As shown in block 54, if the last byte to be sent in the burst has been sent, then the micro-controller 16 goes into a sleep or dormant mode (see block 55) for the predetermined time interval. However, if the last byte has not been sent, then the micro-controller 16 fetches the next byte as shown in block 34.

Referring in particular to Figure 2 of the drawings, the receiver 12 includes a receiver processor 56 defined by a receiver micro-controller 58, for example, a PIC 16F84 or the like. Selected I/O ports of the receiver micro-controller 58 are connected to I/O terminals 60 to allow resident software to be programmed into the micro-controller 58 by an external device such as a PC or the like. The receiver micro-controller 58 is connected via line 62 to the repeater transmitter 14 which is substantially similar to the transmitter circuitry 10. The repeater transmitter 14 includes a transistor 64 which is selectively switched on and off by the receiver micro-controller 58 to generate "1"s and "0"s thereby repeating a signal received by the receiver 12 to the central monitoring or control unit. As in the case of the transmitter circuitry 26, the transmitter 14 sources its operational power directly from a

modulation control signal provided via an output port of the micro-controller 58 via line 62.

The identification signal transmitted by the tag 10 is received by receiver circuitry 68 which has its output signal fed into an RF amplifier 74 and into demodulation circuitry 70 via line 72. The demodulation circuitry 70 has its output connected, via a decoupling capacitor 76, to amplification circuitry generally indicated by reference numeral 78 which comprises a series of operational amplifiers. An output stage of the operational amplifiers is connected via line 80 to a port of the receiver micro-controller 58. Power to the various components of the receiver 12 is provided by a power supply unit 82.

Selected I/O ports of the receiver micro-controller 58 are connected to a programming terminal 84 in which appropriate software to control the method of operation of the receiver 12 is fed into the receiver micro-controller 58. The method of controlling the receiver 12 is set out in Table 1 and described in more detail below.

Referring in particular to Figure 5 of the drawings, the receiver software first includes the method of resetting the bit/byte information as shown in block 86. The initialization of the bit/byte information and various other operating parameters is generally indicated by arrow 160 and following in Table 1. Arrow 162 marks the start of the routine where the ports of the micro-controller 58 are initialized. The software implementing the method then waits for a high input as shown in block 88. When a high input is received, the duration of the pulse is monitored. In particular, the routine "HIG1" (see arrow 164 in Table 1) determines the duration for which the pulse is high and the routine

"HIG2" determines the duration for which the pulse is low (see arrow 166). The sum of the high and low durations is then calculated to check if the total duration is within an acceptable range which is typically between about 50 and about 70 microseconds. Thus, the time duration or interval between the pulses is counted or determined until a next high is received as shown in block 90. If the interval is 10 microseconds (i.e. a five microsecond high followed by a five microsecond low) as shown in block 92, then the incoming bit of the burst of data received from the tag 10 is a "1" as shown in block 94. Thereafter the bit counter is incremented by one as shown in block 96 and if it is the last bit of the count (see block 98) then the bit/byte is reset as shown by line 100 leading into block 86. If, however, the interval is not equal to 10 microseconds, then the interval is timed to determine whether or not it is equal to about 20 microseconds (i.e. 10 microsecond high followed by a 10 microsecond low). If the delay is equal to about 20 microseconds, then the bit is recognized as a "0" as shown in block 104 and, once again, the bit counter is incremented as shown in block 96. However, if the interval is not equal to 20 microseconds then the method includes resetting the bits/byte as shown by line 106 leading to the block 86. The routine for recognizing a "1" or a "0" is generally indicated by arrow 168 in Table 1. Table 1 also includes various other routines, e.g. an RS 232 routine, for feeding data to other devices. It is to be appreciated that the receiver micro-controller 58 may include a variety of additional routines to allow communication of data received from the tag 10 to be communicated to other devices.

The inventor believes that the invention, as illustrated, provides an identification system including a method of communicating data from the tag 10 to the receiver 12 which has reduced power

consumption characteristics. The features of the invention that enhance the low power consumption characteristics include the powering of the transmitter circuitry 26 by means of the modulation control signal and the arrangement in which the state of a bit is determined by the "off" time of the transmitter circuitry 26. Power consumption of the tag 10 is also substantially reduced when the tag 10 is in its sleep or dormant mode.

TABLE 1

```
; ds5000 dev system
;
; capetown with dave
; update check for freq and stat less than 64
; current program
; START SEQUENCE CORRECTED FOR ERRORS AND SPEED INCREASE
; BIT 1 OF DATA CORRECTED
; DECREASE START FRAME HIGH AND LOW
; ADD CHECK FOR CHECKSUM
```

list p = 16f84,f = inhx8m

__CONFIG 3FF1H

INDIR = 0
FSR = 4
PORTA = 5
PORTB = 6
TRISA = 85h
TRISB = 86h
TMRO = 1
STATUS = 3
PCL = 2
OPTN = 81h
INTCON = 0B

RPO	EQU	5	
TEMP2	EQU	11h	
TEMP1	EQU	12h	
TIMER1	EQU	13h	;TIMER VALUE TEMP
VALUEX	EQU	14h	;TEMP
DIGIT1	EQU	15h	;TEMP
DIGIT2	EQU	16h	;TEMP
DIGIT3	EQU	17h	;TEMP
ASCII	EQU	18h	;TEMP
VALUE	EQU	19h	
TEMP	equ	20h	;Temporary storage location
CHAR	EQU	21h	;Character storage location

DIGITS	EQU	22h
VALUE1	EQU	23h
VALUE2	EQU	24h
VALUE3	EQU	25h
VALUE4	EQU	26h
VALUE5	EQU	27h
VALUE6	EQU	28h
VALUE7	EQU	29h
VALUE8	EQU	2Ah
VALUE9	EQU	2Bh
VALUE10	EQU	2Ch
VALUE11	EQU	2Dh
VALUE12	EQU	2Eh
VALUE13	EQU	2Fh
VALUE14	EQU	30h
CALB	EQU	31h
CALA	EQU	32h
DLINE	EQU	33h



```

CHECK1      EQU    36h
CHECK2      EQU    37h
CHECK3      EQU    38h
CHECKX      EQU    39h
CSUM        EQU    3Ah          ;CHECK SUM
CSUM2 EQU    3Bh

```

```

DAT         EQU    PORTB
CNTRL       EQU    PORTA
E           EQU    3
RW          EQU    2
RS          EQU    1
C           EQU    0
W           EQU    0
F           EQU    1
P           EQU    0

```

START

```
CLRF    DLINE
```

```
CLRF    CHECK1
```

```

;      call    LCDInit
;      call    LCDInit
;      CALL    LCDInit

```

```
BCF     PORTB,0
```

;MAIN

;SETUP TIMER/COUNTER

```

movlw    OPTN
movwf    FSR
movlw    00h          ;20h = counter + no prescaler
movwf    INDIR
clr      TMRO

```

```

bcf      STATUS,RPO
clrf     PORTA
clrf     PORTB

```

;Configure ports A and D to outputs

```

bsf      STATUS, RPO          ;Select Register page 1
movlw    B'00000000'          ;Set lower 4 bits in PORTB
movwf    TRISB                ;as outputs
movlw    B'00000011'          ;Set port_a as outputs
movwf    TRISA
bcf      STATUS, RPO          ;Select Register page 0
clrf     PORTA                ;clear port_a

```

```

MOVLW    .5
MOVWF    TEMP2
MOVLW    .10
MOVWF    TIMER1
MOVLW    .100
MOVWF    TEMP1

```

162



20

ZXC	INCF INCF INCF CALL DECFSZ GOTO	TEMP1,F TEMP1,F TEMP1,F BEEP TEMP2,1 ZXC	
	CLRF	PORTA	
MAIN			
	BCF BCF BCF NOP CLRF	PORTB,1 PORTB,4 PORTA,0 CSUM	;B O
SSA			
;	BTFSS	CHECK1,1	
;	GOTO	HIG1	
;	BSF	PORTB,4	;TO OPEN DOOR UN REMARK
;	MOVLW	.5;WAS 20	
;	MOVWF	TEMP2	
;	MOVLW	.10	
;	MOVWF	TIMER1	
;	MOVLW	.100	
;	MOVWF	TEMP1	
;XXC			
;	CALL	BEEP	
;	DECFSZ	TEMP2,1	
;	GOTO	XXC	
;	MOVLW	.50	;100 = 3sec
;	MOVWF	TIMER1	
;RTR1	MOVLW	.200	
;	MOVWF	CHECK2	
;RTR	MOVLW	.200	
;	MOVWF	TEMP1	
;RTR2	DECFSZ	TEMP1,F	
;	GOTO	RTR2	
;	DECFSZ	CHECK2	
;	GOTO	RTR	
;	DECFSZ	TIMER1	
;	GOTO	RTR1	
;	BCF	PORTB,4	
	CLRF	CHECK1	
	CLRF	CHECKX	
HIG1			
	BTFSS	PORTA,0	
	GOTO	HIG1	
	CLRF	TEMP1	
HWW1	NOP		


 164

```

      NOP
      NOP
      INCFSZ TEMP1,F
      GOTO HWZ1
      GOTO HIG1
HWZ1  BTFSC  PORTA,0
      GOTO HWW1      ;wait for HIGH
      BTFSC  TEMP1,3      ;8
      GOTO HIG2
      BTFSC  TEMP1,4      ;16
      GOTO HIG2
      BTFSC  TEMP1,5      ;32
      GOTO HIG2
      BTFSC  TEMP1,6      ;64
      GOTO HIG2
      BTFSC  TEMP1,7      ;128
      GOTO HIG2
      GOTO HIG1

HIG2  CLRFB  TEMP1
HWW2  NOP
      NOP
      NOP
      INCFSZ TEMP1,F
      GOTO HWZ2
      GOTO HIG1
HWZ2  BTFSS  PORTA,0
      GOTO HWW2      ;wait for HIGH
      BTFSC  TEMP1,3      ;8
      GOTO HIG3
      BTFSC  TEMP1,4      ;16
      GOTO HIG3
      BTFSC  TEMP1,5      ;32
      GOTO HIG3
      BTFSC  TEMP1,6      ;64
      GOTO HIG3
      BTFSC  TEMP1,7      ;128
      GOTO HIG3
      GOTO HIG1

HIG3  CLRFB  TEMP1
HWW3  NOP
      NOP
      NOP
      INCFSZ TEMP1,F
      GOTO HWZ3
      GOTO HIG1
HWZ3  BTFSC  PORTA,0
      GOTO HWW3      ;wait for HIGH
      BTFSC  TEMP1,3      ;8
      GOTO HIG4
      BTFSC  TEMP1,4      ;16
      GOTO HIG4
      BTFSC  TEMP1,5      ;32
      GOTO HIG4
      BTFSC  TEMP1,6      ;64
      GOTO HIG4
      BTFSC  TEMP1,7      ;128
      GOTO HIG4
      GOTO HIG1

```

166

HIG4	CLRF	TEMP1	
HWW4	NOP		
	NOP		
	NOP		
	INCF SZ	TEMP1,F	
	GOTO	HWZ4	
	GOTO	HIG1	
HWZ4	BTFSS	PORTA,0	
	GOTO	HWW4	;wait for HIGH
;	BTFSC	TEMP1,3	;8
;	GOTO	HIG5	
	BTFSC	TEMP1,4	;16
	GOTO	HIG5	
	BTFSC	TEMP1,5	;32
	GOTO	HIG5	
	BTFSC	TEMP1,6	;64
	GOTO	HIG5	
	BTFSC	TEMP1,7	;128
	GOTO	HIG5	
	GOTO	HIG1	
HIG5	CLRF	TEMP1	
HWW5	NOP		
	NOP		
	NOP		
	INCF SZ	TEMP1,F	
	GOTO	HWZ5	
	GOTO	HIG1	
HWZ5	BTFSS	PORTA,0	
	GOTO	HWW5	;wait for HIGH
;	BTFSC	TEMP1,8	;8
;	GOTO	HIG6	REPETER ONLY
	BTFSC	TEMP1,4	;16
	GOTO	HIG6	
	BTFSC	TEMP1,5	;32
	GOTO	HIG6	
	BTFSC	TEMP1,6	;64
	GOTO	HIG6	
	BTFSC	TEMP1,7	;128
	GOTO	HIG6	
	GOTO	HIG1	
HIG6	CLRF	TEMP1	
HWW6	NOP		
	NOP		
	NOP		
	INCF SZ	TEMP1,F	
	GOTO	HWZ6	
	GOTO	HIG1	
HWZ6	BTFSS	PORTA,0	
	GOTO	HWW6	;wait for HIGH
;	BTFSC	TEMP1,3	;8
;	GOTO	HIG7	
	BTFSC	TEMP1,4	;16
	GOTO	HIG7	
	BTFSC	TEMP1,5	;32
	GOTO	HIG7	
	BTFSC	TEMP1,6	;64
	GOTO	HIG7	

BTFSC	TEMP1,7	
GOTO	HIG7	;128
GOTO	HIG1	

HIG7

;*****END*****

```

;          CLRf          VALUE1
;SX1      BTFSC          PORTA,0
;          GOTO          SX1
;          INCf          VALUE1
;          MOVLW         .200
;          MOVWF         VALUE2
;LX1      DECFSZ         VALUE2,F
;          GOTO          NOE
;          GOTO          ENDF
;NOE      BTFSS          PORTA,0
;          GOTO          LX1
;          GOTO          SX1

;ENDF     GOTO          SKIPP

;          NOP

;          CLRf          VALUE1

          CALL          WAITD
          MOVF          TIMER1,W
          MOVWF         VALUE1

          CALL          WAITD
          MOVF          TIMER1,W
          MOVWF         VALUE2

          CALL          WAITD
          MOVF          TIMER1,W
          MOVWF         VALUE3

          CALL          WAITD
          MOVF          TIMER1,W
          MOVWF         VALUE4

          CALL          WAITD
          MOVF          TIMER1,W
          MOVWF         VALUE5
          CALL          WAITD
          MOVF          TIMER1,W
          MOVWF         VALUE6

          CALL          WAITD
          MOVF          TIMER1,W
          MOVWF         VALUE7

```

	CALL	WAITD	
	MOVF	TIMER1,W	
	MOVWF	VALUE8	
	CALL	WAITD	
	MOVF	TIMER1,W	
	MOVWF	VALUE9	
	CALL	WAITD	
	MOVF	TIMER1,W	
	MOVWF	VALUE10	
	CALL	WAITD	
	MOVF	TIMER1,W	
	MOVWF	VALUE11	
	CALL	WAITD	
	MOVF	TIMER1,W	
	MOVWF	VALUE12	
	CALL	WAITD	
	MOVF	TIMER1,W	
	MOVWF	VALUE13	
	CALL	WAITD	
	MOVF	TIMER1,W	
	MOVWF	VALUE14	
	CALL	WAITD	
	MOVF	TIMER1,W	
	MOVWF	VALUE15	
	CALL	WAITD	
	MOVF	TIMER1,W	
	MOVWF	VALUE16	
	CALL	WAITD	
	MOVF	TIMER1,W	
	MOVWF	CSUM	;CHECK SUM
;	MOVLW	.98	
;	MOVWF	VALUE15	;REC ID
;	;CHECK DATA		
;	goto	tst7	
	MOVF	VALUE1,W	
	ADDWF	VALUE2,W	
	ADDWF	VALUE3,W	
	ADDWF	VALUE4,W	
	ADDWF	VALUE5,W	
	ADDWF	VALUE6,W	
	ADDWF	VALUE7,W	
	ADDWF	VALUE8,W	
	ADDWF	VALUE9,W	
	ADDWF	VALUE10,W	
	ADDWF	VALUE11,W	
	ADDWF	VALUE12,W	
	ADDWF	VALUE13,W	

```

ADDWF      VALUE14,W
ADDWF      VALUE15,W
MOVWF      CSUM2

MOVF       CSUM,W          ;CHECK SUM
SUBWF      CSUM2,W
MOVWF      TEMP1
BTFSC      TEMP1,0        ;
GOTO       OUT            ;
BTFSC      TEMP1,1        ;
GOTO       OUT            ;
BTFSC      TEMP1,2
GOTO       OUT
BTFSC      TEMP1,3
GOTO       OUT
BTFSC      TEMP1,4
GOTO       OUT
BTFSC      TEMP1,5
GOTO       OUT
BTFSC      TEMP1,6
GOTO       OUT
BTFSC      TEMP1,7
GOTO       OUT

MOVF       CSUM2,W          ;CHECK SUM
SUBWF      CSUM,W
MOVWF      TEMP1
BTFSC      TEMP1,0        ;
GOTO       OUT            ;
BTFSC      TEMP1,1        ;
GOTO       OUT            ;
BTFSC      TEMP1,2
GOTO       OUT
BTFSC      TEMP1,3
GOTO       OUT
BTFSC      TEMP1,4
GOTO       OUT
BTFSC      TEMP1,5
GOTO       OUT
BTFSC      TEMP1,6
GOTO       OUT
BTFSC      TEMP1,7
GOTO       OUT

MOVF       VALUE2,W        ;CHECK STAT AND FREQ
MOVWF      TEMP1
BTFSC      TEMP1,6
GOTO       OUT
BTFSC      TEMP1,7
GOTO       OUT
MOVF       VALUE3,W        ;CHECK STA AND FREQ
MOVWF      TEMP1
BTFSC      TEMP1,6
GOTO       OUT
BTFSC      TEMP1,7
GOTO       OUT

```

```

;          goto          tsp7          ;skip looking for P or Q

;look for P OR Q

MOVLW      .80                      ;was 80h less blt0
SUBWF VALUE 16,W
MOVWF      TEMP1
;          BTFSC         TEMP1,0      ;LOOK FOR P ONLY
;          GOTO          OUT          ;REM =
;          BTFSC         TEMP1,1      ;P,Q,R,S
;          GOTO          OUT          ;
;          BTFSC         TEMP1,2
;          GOTO          OUT
;          BTFSC         TEMP1,3
;          GOTO          OUT
;          BTFSC         TEMP1,4
;          GOTO          OUT
;          BTFSC         TEMP1,5
;          GOTO          OUT
;          BTFSC         TEMP1,6
;          GOTO          OUT
;          BTFSC         TEMP1,7
;          GOTO          OUT

```

tst7

;TURN ON 10 PORTB 0 PIN 6

```

;          MOVW          .84          ;T
;          SUBWF VALUE1,W
;          MOVWF         TEMP1
;          BTFSC         TEMP1,0      ;remark
;          GOTO          KKK          ;WAS KKL
;          BTFSC         TEMP1,1
;          GOTO          KKK
;          BTFSC         TEMP1,2
;          GOTO          KKK
;          BTFSC         TEMP1,3
;          GOTO          KKK
;          BTFSC         TEMP1,4
;          GOTO          KKK
;          BTFSC         TEMP1,5
;          GOTO          KKK
;          BTFSC         TEMP1,6
;          GOTO          KKK
;          BTFSC         TEMP1,7
;          GOTO          KKK
;          INCF          CHECK1
;          GOTO          KKD

```

```

KKK        MOVLW          .69          ;E
;          SUBWF VALUE1,W
;          MOVWF         TEMP1
;          BTFSC         TEMP1,0      ;remark
;          GOTO          KKS          ;WAS KKL
;          BTFSC         TEMP1,1
;          GOTO          KKS
;          BTFSC         TEMP1,2
;          GOTO          KKS
;          BTFSC         TEMP1,3

```

```

;      GOTO      KKS
;      BTFSC     TEMP1,4
;      GOTO      KKS
;      BTFSC     TEMP1,5
;      GOTO      KKS
;      BTFSC     TEMP1,6
;      GOTO      KKS
;      BTFSC     TEMP1,7
;      GOTO      KKS
;      INCF      CHECKX
;      GOTO      KKD

KKS      CLRF      CHECK1
          CLRF      CHECKX

KKD

;      BSF       PORTB,4
;      MOVLW     .5
;      MOVWF     TEMP2
;      MOVLW     .20
;      MOVWF     TIMER1
;      MOVLW     .200
;      MOVWF     TEMP1
;      MOVLW     .50      ;100 = 3sec
;      MOVWF     TIMER1
;RTR1     MOVLW     .200
;          MOVWF     CHECK2
;RTR      MOVLW     .200
;          MOVWF     TEMP1
;RTR2     DECFSZ    TEMP1,F
;          GOTO      RTR2
;          DECFSZ    CHECK2
;          GOTO      RTR
;          DECFSZ    TIMER1
;          GOTO      RTR1

KKL

          ;check for alarm on

OKD
;      MOVLW     .50      ;was 49
;      SUBWF     VALUE3,W
;      MOVWF     TEMP1
;      BTFSC     TEMP1,0
;      GOTO      OUT3 ;WAS 1
;      BTFSC     TEMP1,1
;      GOTO      OUT3
;      BTFSC     TEMP1,2
;      GOTO      OUT3
;      BTFSC     TEMP1,3
;      GOTO      OUT3
;      BTFSC     TEMP1,4
;      GOTO      OUT3
;      BTFSC     TEMP1,5
;      GOTO      OUT3
;      BTFSC     TEMP1,6
;      GOTO      OUT3

```

```

;          BTFSC      TEMP1,7
;          GOTO       OUT3

;          CLRF        CHECK1
;          GOTO       OUT2 ;WAS 1

OUT2      ;ALARM ON BEEP

;          MOVLW       .5
;          MOVWF       TEMP2
;          MOVLW       .10
;          MOVWF       TIMER1
;          MOVLW       .100
;          MOVWF       TEMP1

;ZXC1 INCF      TEMP1,F
;          INCF        TEMP1,F
;          INCF        TEMP1,F
;          CALL        BEEP
;          DECFSZ      TEMP2,1
;          GOTO       ZXC1

;          MOVLW       .200
;          MOVWF       TIMER1
;          MOVLW       .100
;          MOVWF       TEMP1
;          CALL        BEEP
;          GOTO       SKIPP

OUT1      ;NOT SEEN A T

;SEEN UNIT NOT A T SO SET OFF ALARM IF SEEN MORE THAN 4 TIMES

;          BTFSS      CHECK1,3      ;WAS 3
;          INCF        CHECK1,F
;          BTFSS      CHECK1,3
;          GOTO OUT3
;          NOP
;          GOTO OUT2      ;SET OFF ALARM

OUT3      ;
;          MOVLW       .10          ;was a 5 TIC
;          MOVWF       TIMER1
;          MOVLW       .20          ;was a 10
;          MOVWF       TEMP1
;          CALL        BEEP
;          GOTO       SKIP

OUT       ;VALUE NOT RIGHT RETURN AND READ AGAIN

;          MOVLW       .200
;          MOVWF       TIMER1
;          MOVLW       .100

```

```

;      MOVWF      TEMP1
;      CALL       BEEP

      GOTO        MAIN

SKIPP

OUTV

      BSF         PORTB,1
      GOTO        GKL          ;FOR NO DISPLAY

GKL                                ;SEND ONLY ALARM DATA

      BTFSS       PORTA,1
      GOTO        TX2

      MOVLW       .50          ;was 2A=T
      SUBWF       VALUE3,W
      MOVWF       TEMP1
      BTFSC       TEMP1,0
      GOTO        TX1
      BTFSC       TEMP1,1
      GOTO        TX1
      BTFSC       TEMP1,2
      GOTO        TX1
      BTFSC       TEMP1,3
      GOTO        TX1
      BTFSC       TEMP1,4
      GOTO        TX1
      BTFSC       TEMP1,5
      GOTO        TX1
      BTFSC       TEMP1,6
      GOTO        TX1
      BTFSC       TEMP1,7
      GOTO        TX1
      GOTO        TX2

TX1      GOTO      MAIN

TX2

      ;PC VERSION FOR DS5000

      MOVLW       65h
      CALL        TXDATA
      MOVLW       65h
      CALL        TXDATA

      MOVLW       .33
      CALL        TXDATA
      MOVLW       .42
      CALL        TXDATA

```

```

;
;
        MOVLW      .42      ;*
        CALL      TXDATA
        MOVF      VALUE1,W
        CALL      TXDATA
        MOVF      VALUE2,W
        CALL      TXDATA
        MOVF      VALUE3,W
        CALL      TXDATA

        MOVLW      .65
        CALL      TXDATA
        MOVLW      .66
        CALL      TXDATA
        MOVLW      .67
        CALL      TXDATA

        MOVF      VALUE4,W
        CALL      TXDATA

;
        MOVF      VALUE5,W
        MOVLW      .48
        CALL      TXDATA
        MOVF      VALUE6,W
        CALL      TXDATA
        MOVF      VALUE7,W
        CALL      TXDATA
        MOVF      VALUE8,W
        CALL      TXDATA

        MOVF      VALUE1,W
        CALL      TXDATA
        MOVF      VALUE9,W
        CALL      TXDATA
        MOVF      VALUE10,W
        CALL      TXDATA
        MOVF      VALUE11,W
        CALL      TXDATA
        MOVF      VALUE12,W
        CALL      TXDATA
        MOVF      VALUE13,W
        CALL      TXDATA
        MOVF      VALUE14,W
        CALL      TXDATA
        MOVF      VALUE15,W
        CALL      TXDATA
        MOVF      VALUE16,W
        CALL      TXDATA

        MOVLW      .48
        CALL      TXDATA

        MOVLW      0Ah
        CALL      TXDATA
        MOVLW      0Dh
        CALL      TXDATA
        MOVLW      80h
        CALL      TXDATA

        BSF        PORTB,2

```

TERRYX

```

MOV LW    .01
MOV WF    CHECK3

```

TERRY Y

```

;
; GOTO     MAIN
; GOTO     NOTX

```

;tx data on

```

MOV F     VALUE1,0
ADD WF    VALUE2,0
ADD WF    VALUE3,0
ADD WF    VALUE4,0
ADD WF    VALUE5,0
ADD WF    VALUE6,0
ADD WF    VALUE7,0
ADD WF    VALUE8,0
ADD WF    VALUE9,0
ADD WF    VALUE10,0
ADD WF    VALUE11,0
ADD WF    VALUE12,0
ADD WF    VALUE13,0
ADD WF    VALUE14,0
ADD WF    VALUE15,0

```

```

MOV WF    CSUM

```

```

MOV LW    20h    ;was 20
MOV WF    TEMP1

```

```

movlw 20h
movwf TEMP
CALL   pulse4

```

```

MOV LW    20h
MOV WF    TEMP1

```

```

MOV LW    20h    ;WAS 20 AND START
MOV WF    TEMP
CALL   pulse4

```

```

movlw 20h
movwf TEMP
call   pulse4

```

```

MOV LW    .04    ;TX ON TIME
MOV WF    TEMP1

```

```

movf     VALUE 1,0    ;was 10h
movwf    TEMP
call     pulse

```

```

movf     VALUE2,0
movwf    TEMP
call     pulse

```

```
;      movlw      .50
      movf      VALUE3,0
      movwf     TEMP
      call      pulse

      movf      VALUE4,0
      movwf     TEMP
      call      pulse

      movf      VALUE5,0
      movwf     TEMP
      call      pulse

      movf      VALUE6,0
      movwf     TEMP
      call      pulse

      movf      VALUE7,0
      movwf     TEMP
      call      pulse

      movf      VALUE8,0
      movwf     TEMP
      call      pulse

      movf      VALUE9,0
      movwf     TEMP
      call      pulse

      movf      VALUE10,0
      movwf     TEMP
      call      pulse

      movf      VALUE11,0
      movwf     TEMP
      call      pulse

      movf      VALUE12,0
      movwf     TEMP
      call      pulse

      movf      VALUE13,0
      movwf     TEMP
      call      pulse

      movf      VALUE14,0
      movwf     TEMP
      call      pulse

      movf      VALUE15,0
      movwf     TEMP
      call      pulse

;      movf      VALUE16,0
      movlw      .83
      movwf     TEMP
      call      pulse

      MOVF      CSUM,0
```

MOVWF TEMP
CALL pulse

call pulse5

DECFSZ CHECK3,F
GOTO TERRY

NOTX

MOVLW 65h
CALL XXDATA

MOVLW 65h
CALL XXDATA

MOVLW .33 ;!

CALL XXDATA

MOVLW .42 ;*

CALL XXDATA

MOVLW .42 ;*

CALL XXDATA

; MOVF VALUE1,W

; CALL XXDATA

MOVF VALUE2,W

CALL XXDATA

MOVF VALUE3,W

CALL XXDATA

MOVLW .65

CALL XXDATA

MOVLW .66

CALL XXDATA

MOVLW .67

CALL XXDATA

MOVF VALUE4,W

CALL XXDATA

MOVF VALUE5,W

MOVLW .48

CALL XXDATA

MOVF VALUE6,W

CALL XXDATA

MOVF VALUE7,W

CALL XXDATA

MOVF VALUE8,W

CALL XXDATA

MOVF VALUE1,W

CALL XXDATA

MOVF VALUE9,W

CALL XXDATA

MOVF VALUE10,W

CALL XXDATA

MOVF VALUE11,W

CALL XXDATA

MOVF VALUE12,W

CALL XXDATA

MOVF VALUE13,W

```

CALL    XXDATA
MOVF    VALUE14,W
CALL    XXDATA
MOVF    VALUE15,W
CALL    XXDATA
MOVF    VALUE16,W
CALL    XXDATA
MOVLW   .48
CALL    XXDATA

```

```

MOVLW   0Ah
CALL    XXDATA
MOVLW   0Dh
CALL    XXDATA
MOVLW   80h
CALL    XXDATA
MOVLW   80h
CALL    XXDATA

```

```
BCF     PORTB,2
```

```
GOTO    MAIN
```

pulse

```

BTFSS   TEMP,0
CALL    pulse3
BTFSC   TEMP,0
call    pulse2

```

```

BTFSS   TEMP,1
CALL    pulse3
BTFSC   TEMP,1
call    pulse2

```

```

BTFSS   TEMP,2
CALL    pulse3
BTFSC   TEMP,2
call    pulse2

```

```

BTFSS   TEMP,3
CALL    pulse3
BTFSC   TEMP,3
call    pulse2

```

```

BTFSS   TEMP,4
CALL    pulse3
BTFSC   TEMP,4
call    pulse2

```

```

BTFSS   TEMP,5
CALL    pulse3
BTFSC   TEMP,5
call    pulse2

```

```

        BTFSS      TEMP,6
        CALL      pulse3
        BTFSC      TEMP,6
        call      pulse2

        BTFSS      TEMP,7
        CALL      pulse5
        BTFSC      TEMP,7
        call      pulse6
        CALL      pulse5
;
return
pulse6  clrwdt

        MOVF      TEMP1,W
        movwf     TIMER1
        bsf       PORTB,3      ;on foR 3 6 9 12 ETC.
diiig   NOP
        NOP
        NOP

        nop
        nop
        nop

        nop      ;new
        nop
        nop
        nop

        decfsz    TIMER1,1
        goto      diiig
        BCF       PORTB,3      ;cycle 46 turn off

        MOVLW     .22
        MOVWF     TIMER1
        TNY2      DECFSZ    TIMER1,1
        GOTO      TNY2

        RETURN

pulse2  clrwdt

        MOVF      TEMP1,W
        movwf     TIMER1
        bsf       PORTB,3      ;on foR 3 6 9 12 ETC.
ddgfg   NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP

```

36

```

nop
nop
nop

decfsz    TIMER1,1
goto      ddgf
BCF       PORTB,3      ;cycl 46 turn off

MOV LW    .17          ;WAS 13
MOVWF     TIMER1
TNY1 DECFSZ TIMER1,1
      GOTO   TNY1

RETURN

pulse5 clrwdt

      MOVF    TEMP1,W
      movwf   TIMER1
      bsf     PORTB,3      ;on foR 3 6 9 12 ETC.
digg NOP
      NOP
      NOP
      decfsz   TIMER1,1
      goto     digg
      BCF      PORTB,3      ;cycle 46 turn off
      NOP
      NOP
      NOP
      NOP
      NOP
      NOP

RETURN

pulse 3 clrwdt

      MOVF    TEMP1,W
      movwf   TIMER1
      bsf     PORTB,3      ;on foR 3 6 9 12 ETC.
digpp NOP
      NOP
      NOP
      decfsz   TIMER1,1
      goto     digpp
      BCF      PORTB,3      ;cycle 46 turn off

      MOV LW    .3
      MOVWF     TIMER1
TNY4 NOP
      NOP
      NOP
      DECFSZ    TIMER1,1
      GOTO      TNY4

```

RETURN

```

pulse4  clrwdt
        MOVF      TEMP1,W
        movwf     TIMER1
        NOP
        NOP      ;12 CYCLES
        NOP
        NOP
        bsf       PORTB,3      ;on for 46us
digph   NOP
        NOP
        NOP
        decfsz    TIMER1,1
        goto      digph
digf    BCF       PORTB,3      ;cycle 46 turn off
        NOP
        NOP
        NOP
        decfsz    TEMP,1
        goto      digf
        NOP
        NOP
        RETURN

```

TXDATA

```

        MOVWF     TEMP

        call      PPSE2
        nop
        nop
        nop
        nop
        nop
        nop

        BTFSS     TEMP,0
        CALL      PPSE2
        BTFSC     TEMP,0
        call      PPSE3

        BTFSS     TEMP,1
        CALL      PPSE2
        BTFSC     TEMP,1
        call      PPSE3

        BTFSS     TEMP,2
        CALL      PPSE2
        BTFSC     TEMP,2
        call      PPSE3

        BTFSS     TEMP,3
        CALL      PPSE2
        BTFSC     TEMP,3
        call      PPSE3

```

BTFSS	TEMP,4	
CALL	PPSE2	
BTFSC	TEMP,4	
call	PPSE3	
BTFSS	TEMP,5	
CALL	PPSE2	
BTFSC	TEMP,5	
call	PPSE3	
BTFSS	TEMP,6	
CALL	PPSE2	
BTFSC	TEMP,6	
call	PPSE3	
BTFSS	TEMP,7	
CALL	PPSE2	
BTFSC	TEMP,7	
call	PPSE3	
CALL	PPSE3	
call	PPSE3	
return		
XXDATA		
MOVWF	TEMP	
;	call	PPSE3
;	call	PPSE3
;	call	PPSE3
;	call	PPSE3
;	call	PPSE3
call	XPSE2	;was2
nop		
nop		
nop		
nop		
nop		
BTFSS	TEMP,0	
CALL	XPSE2	
BTFSC	TEMP,0	
call	XPSE3	
BTFSS	TEMP,1	
CALL	XPSE2	
BTFSC	TEMP,1	
call	XPSE3	
BTFSS	TEMP,2	
CALL	XPSE2	
BTFSC	TEMP,2	
call	XPSE3	
BTFSS	TEMP,3	
CALL	XPSE2	

	BTFSC call	TEMP,3 XPSE3	
	BTFSS CALL	TEMP,4 XPSE2	
	BTFSC call	TEMP,4 XPSE3	
	BTFSS CALL	TEMP,5 XPSE2	
	BTFSC call	TEMP,5 XPSE3	
	BTFSS CALL	TEMP,6 XPSE2	
	BTFSC call	TEMP,6 XPSE3	
	BTFSS CALL	TEMP,7 XPSE2	
	BTFSC call	TEMP,7 XPSE3	
	CALL call	XPSE3 XPSE3	;was3
;	CALL	XPSE2	
return			
PPSE2	clrwdt		
	MOVLW	.7	
	movwf	TEMP1	
	bsf	PORTB,2	;on foR 3 6 9 12 ETC.
dighh	decfsz	TEMP1,1	
	goto	dighh	
	NOP		
	NOP		
	NOP		
RETURN			
PPSE3	clrwdt		
	MOVLW	.7	
	movwf	TEMP1	
	bcf	PORTB,2	;on foR 3 6 9 12 ETC.
diggh	decfsz	TEMP1,1	
	goto	diggh	
	NOP		
	NOP		
	NOP		
RETURN			
XPSE2	clrwdt		
	MOVLW	.7	
	movwf	TEMP1	
	bcf	PORTB,2	;was bsf on foR 3 6 9 12 ETC.

Xighh	decfsz	TEMP1,1	
	goto	Xighh	
	NOP		
	NOP		
	NOP		
RETURN			
XPSE3		clrwdt	
	MOVLW	.7	
	movwf	TEMP1	
	bsf	PORTB,2	;was bcf on for 3 6 9 12 ETC.
Xiggh	decfsz	TEMP1,1	
	goto	Xiggh	
	NOP		
	NOP		
	NOP		
RETURN			
BEEP			
DS2	movf	TEMP1,W	
	movwf	TEMP	
	BSF	PORTB,1	;WAS1
SD2	decfsz	TEMP,F	
	goto	SD2	
	BCF	PORTB,1	
	movf	TEMP1,W	
	movwf	TEMP	
SD3	decfsz	TEMP,F	
	goto	SD3	
	DECFSZ	TIMER1,F	
	GOTO	DS2	
	return		
WAITD			
	CLRF	TIMER1	
	CALL	WAITDD	
	BTFSC	TEMP1,3	
	BSF	TIMER1,0	
	CALL	WAITDD	
	BTFSC	TEMP1,3	
	BSF	TIMER1,1	
	CALL	WAITDD	
	BTFSC	TEMP1,3	
	BSF	TIMER1,2	
	CALL	WAITDD	
	BTFSC	TEMP1,3	
	BSF	TIMER1,3	
	CALL	WAITDD	
	BTFSC	TEMP1,3	
	BSF	TIMER1,4	
	CALL	WAITDD	
	BTFSC	TEMP1,3	
	BSF	TIMER1,5	

168

CALL	WAITDD
BTFSC	TEMP1,3
BSF	TIMER1,6
CALL	WAITDD
BTFSC	TEMP1,3
BSF	TIMER1,7

RETURN

WAITDD

CLRF	TEMP1
------	-------

S1	incf	TEMP1,F
;	NOP	
	NOP	
	NOP	
	NOP	
	NOP	
	NOP	
	BTFSC	PORTA,0
	GOTO	S1

L1

nop
nop
nop
nop
INCFSZ TEMP1,F
GOTO LL1
GOTO HH1

LL1	BTFSS	PORTA,0
	GOTO	L1

HH1

RETURN

ASC2

;	MOVLW	20h
;	CALL	TXDATA
	MOVLW	.48
	MOVWF	ASCII

RPT4	INCF	DIGIT1
	DECFSZ	ASCII,F
	GOTO	RPT4
	MOVLW	.48
	MOVWF	ASCII
	MOVF	DIGIT1,W
	call	TXDATA
RPT5	INCF	DIGIT2
	DECFSZ	ASCII,F
	GOTO	RPT5

```

        MOVLW      .48
        MOVWF      ASCII
        MOVF       DIGIT2,W
        call       TXDATA
RPT6    INCF       DIGIT3
        DECFSZ     ASCII,F
        GOTO       RPT6
        MOVF       DIGIT3,W
        call       TXDATA

```

```

;      MOVLW      20h
;      CALL       TXDATA

```

RETURN

ASC

```

        MOVLW      .48
        MOVWF      ASCII

RPT1    INCF       DIGIT1
        DECFSZ     ASCII,F
        GOTO       RPT1
        MOVLW      .48
        MOVWF      ASCII
        MOVF       DIGIT1,W
        call       SendChar
RPT2    INCF       DIGIT2
        DECFSZ     ASCII,F
        GOTO       RPT2
        MOVLW      .48
        MOVWF      ASCII
        MOVF       DIGIT2,W
        call       SendChar
RPT3    INCF       DIGIT3
        DECFSZ     ASCII,F
        GOTO       RPT3
        MOVF       DIGIT3,W
        call       SendChar
        RETURN

```

ASC1

```

        MOVWF      DIGIT1
        MOVF       DIGIT1,W
        call       SendChar

```

RETURN

CONVERT

```

        CLRF       DIGIT1
        CLRF       DIGIT2
        CLRF       DIGIT3
        MOVWF      VALUE

```

```

        CALL      D1
        CALL      D2
        CALL      D3
        RETURN

D1      movlw    .100
        subwf    VALUE,W
        BTFSS    STATUS,C
        RETLW    0
        MOVWF    VALUE      ;BALANCE IN VALUEA
        INCF     DIGIT1,F
        GOTO     D1

D2      movlw    .10
        subwf    VALUE,W
        BTFSS    STATUS,C
        RETLW    0
        MOVWF    VALUE      ;BALANCE IN VALUEA
        INCF     DIGIT2,F
        GOTO     D2

D3      movlw    .1
        subwf    VALUE,W
        BTFSS    STATUS,C
        RETLW    0
        MOVWF    VALUE      ;BALANCE IN VALUEA
        INCF     DIGIT3,F
        GOTO     D3

```

```

CLRLCD
    return

```

```

SetupDelay
    return

```

```

SendChar
    return

```

```

SendCmd
    return

```

```

BusyCheck
    return

```

```

LCDInit

```

```

SLP
    return

```

```

END

```

TABLE 2

;T140798

;
;
;
;
;
;current project
ADD START PULSE TO DECREASE TO 48uS
INCREASE START PULSE TO 200uS
SET INPUT TO BE AT CENTRE FRAME 100BAUD

list p= 12C509,f=inhx8m

; _IDLOCS 0000H

_CONFIG 000EH ;000EH for int 4 meg osc*****
;000DH FOR EXT 4 MEG XTAL
;001EH FOR MCLR ON EXT PIN 4 + INT OSC
;001DH FOR MCLR ON EXT PIN 4 TO VSS+EXT XTAL
;000CH FOR EXT 32KHZ

_IDLOCS 0000H

STATUS=3

OSCCAL=5

;12C509

INDIR=0

FSR=4

PORTA=6

;WAS 5

;PORTB=6

;TRISA=85h

;TRISB=86h

;OPTN=81h

TMRO=1

VAL1=08h

;code to send

VAL0=09h

;delay/repeat

DIGIT5=10h

DIGIT6=11h

DIGIT7=12h

DIGIT8=13h

DIGIT2=14h

DIGIT3=15h

DIGIT4=16h

DIGIT1=17h

DIGIT9=18h

DIGIT10=19h

DIGIT11=1Ah

DIGIT12=1Bh

DIGIT13=1Ch

DIGIT14=1Dh

DIGIT15=1Eh

VAL4=1Fh

SPARE=0Fh

TERRY2=0Eh

VAL5=0Dh

VAL2=0Ch

TERRY=0Bh

VAL3=0Ah

```

;      ORG      0
;      MOVWF    OSCCAL

start

      ORG      0
      MOVWF    OSCCAL

      ;SETUP PORT DDR
      MOVLW 3EH
      TRIS PORTA                ;all INPUTS EXEPT FOR IO 1

      BCF      PORTA,0

      MOVLW    004FH            ;WAS OF 4F    DEFALT IS SLOW WAS 00ADH FOR NO
WAKE  OPTION                ;                UP ON PIN CHANGE

;      BTFSS    PORTA,2
;      GOTO     CHECK

      BSF      STATUS,5        ;FIXED SETUP
      CALL     SET_UP
      BCF      STATUS,5

loop

;      INCFSZ   DIGIT8,1
;      GOTO     OOT
;      INCFSZ   DIGIT7,1
;      GOTO     OOT
;      INCFSZ   DIGIT6,1
;      GOTO     OOT
;      INCFSZ   DIGIT5,1
;      GOTO     OOT

OOT    NOP
      BTFSS    TERRY,7        ;IF 0 GOTO X
      GOTO     OUTA
      MOVLW    14h
      MOVWF    TERRY
OUTA   BTFSS    TERRY,6
      GOTO     OUTB
      MOVLW    14h
      MOVWF    TERRY
OUTB   BTFSS    TERRY,5
      GOTO     OUTC

      MOVLW    14h
      MOVWF    TERRY

OUTC   BTFSC    TERRY
      GOTO     ALLOK
      BTFSC    TERRY,6
      GOTO     ALLOK
      BTFSC    TERRY,5

```

```

        GOTO      ALLOK
        BTFSC     TERRY,4
        GOTO      ALLOK
;       BTFSC     TERRY,3
;       GOTO      ALLOK
        MOVLW     14h
        MOVWF     TERRY

```

ALLOK

;*****WAS HERE

```

;       BTFSS     PORTA,2
        GOTO      CHECK

```

;*****TO HERE

NEXT CLRWDI

```

;       BTFSS     PORTA,2      ;*****
;       GOTO      NEXT
        CLRF      PORTA

```

PSD

```

        BSF       STATUS,5      ;FIXED SETUP
        CALL      CSUM
        BCF       STATUS,5

```

;*****CHECK SUM FOR DATA

```

;       MOVLW     .48
;       SUBWF     DIGIT15,W
;       MOVWF     VAL1
;       BTFSC     VAL1,0
;       GOTO      TX1
;       BTFSC     VAL1,1
;       GOTO      TX1
;       BTFSC     VAL1,2
;       GOTO      TX1
;       BTFSC     VAL1,3
;       GOTO      TX1
;       BTFSC     VAL1,4
;       GOTO      TX1
;       BTFSC     VAL1,5
;       GOTO      TX1
;       BTFSC     VAL1,6
;       GOTO      TX1
;       BTFSC     VAL1,7
;       GOTO      TX1

```

```

;       GOTO      TX2

```

```

        MOVLW     20h      ;was 20 AND 11
        MOVWF     VAL5

```

```

        movlw     20h

```

47

movwf	VAL1	
CALL	pulse4	
MOVLW	20h	
MOVWF	VAL5	
MOVLW	20h	;WAS 20 AND START
MOVWF	VAL1	
CALL	pulse4	
movlw	20h	
movwf	VAL1	
call	pulse4	
MOVLW	.04	;was 04 TX ON TIME
MOVWF	VAL5	
movf	DIGIT1,0	;DIGIT1
MOVWF	VAL1	
call	pulse	
movf	DIGIT2,0	;period of tx
movwf	VAL1	
call	pulse	
movf	DIGIT3,0	;+ -1 on io change
movwf	VAL1	
call	pulse	
movf	DIGIT4,0	;inc on io change
movwf	VAL1	
call	pulse	
movf	DIGIT5,0	;inc on tx 1
movwf	VAL1	
call	pulse	
movf	DIGIT6,0	;inc on tx 2
movwf	VAL1	
call	pulse	
movf	DIGIT7,0	;inc on tx 3
movwf	VAL1	
call	pulse	
movf	DIGIT8,0	
movwf	VAL1	
call	pulse	
movf	DIGIT9,0	
movwf	VAL1	
call	pulse	
movf	DIGIT10,0	
movwf	VAL1	
call	pulse	
movf	DIGIT11,0	
movwf	VAL1	
call	pulse	

```

movf      DIGIT12,0
movwf     VAL1
call      pulse

```

```

movf      DIGIT13,0
movwf     VAL1
call      pulse

```

```

movf      DIGIT14,0
movwf     VAL1
call      pulse

```

```

movf      DIGIT15,0
movwf     VAL1
call      pulse

```

```

movlw     .80
movwf     VAL1
call      pulse

```

```

movf      TERRY2,0
movwf     VAL1
call      pulse

```

```

nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop

```

```

call      pulse5

```

```

;SLEEP MODE AFTER CODE

```

```

TX2  clrwdt
      MOVF  PORTA,0
      MOVWF VAL1
      sleep
      nop
      MOVF  VAL1,0
      MOVWF PORTA
      clrwdt

```

```

RETURN

```

```

ZERO  NOP
      MOVWF VAL1
      BTFSS VAL1,7      ;IF 0 GOTO X
      GOTO  OUTX

```

```

OUTX  CLRF      VAL1
      INCF      VAL1,1
      BTFSS     VAL1,6
      GOTO      OUTX2
      CLRF      VAL1
      INCF      VAL1,1
      OUTX2

```

```

RETURN

```

```

;PROC FOR OUTPUT CODE

```

```

pulse

```

```

      BTFSS     VAL1,0
      CALL      pulse3
      BTFSC     VAL1,0
      call      pulse2

```

```

      BTFSS     VAL1,1
      CALL      pulse3
      BTFSC     VAL1,1
      call      pulse2

```

```

      BTFSS     VAL1,2
      CALL      pulse3
      BTFSC     VAL1,2
      call      pulse2

```

```

      BTFSS     VAL1,3
      CALL      pulse3
      BTFSC     VAL1,3
      call      pulse2

```

```

      BTFSS     VAL1,4
      CALL      pulse3
      BTFSC     VAL1,4
      call      pulse2

```

```

      BTFSS     VAL1,5
      CALL      pulse3
      BTFSC     VAL1,5
      call      pulse2

```

```

      BTFSS     VAL1,6
      CALL      pulse3
      BTFSC     VAL1,6
      call      pulse2

```

```

      BTFSS     VAL1,7
      CALL      pulse5
      BTFSC     VAL1,7
      call      pulse6

```

```

;      CALL      pulse5
      return
pulse6 clrwtdt

```

156

50

[illegible]

```

;      nop
      nop
      nop
      nop
      nop
      nop
      nop
      nop
      nop
      nop
      nop
      nop
      nop
      nop
      nop
      nop
      nop
      nop
      nop
      nop
      RETURN

```

pulse5 clrwdt

```

;      MOVLW      .10      ;TEST OF RANGE
      MOVF      VAL5,W
      movwf     VAL0
      bsf       PORTA,0    ;on foR 3 6 9 12 ETC.
digg  decfsz    VAL0,1
      goto     digg
      BCF      PORTA,0    ;cycle 46 turn off
      RETURN

```

pulse3 clrwdt

```

      MOVF      VAL5,W
      movwf     VAL0
      bsf       PORTA,0    ;on foR 3 6 9 12 ETC.
digpp decfsz    VAL0,1
      goto     digpp
      BCF      PORTA,0    ;cycle 46 turn off
      nop
      nop
      nop
      nop
      nop
      nop
      nop
      RETURN

```

pulse4 clrwdt

```

      MOVF      VAL5,W
      movwf     VAL0
      NOP
      NOP      ;12 CYCLES
      bsf       PORTA,0    ;on for 46us

```

152



```

digph  decfsz    VAL0,1
        goto     digph
        BCF      PORTA,0      ;cycle 46 turn off
digf    decfsz    VAL1,1
        goto     digf
        NOP

```

RETURN

;*****MOVED TO HERE

CHECK

;COUNTER MODE

```

        MOVLW     .48
        SUBWF     DIGIT15,W
        MOVWF     VAL1
        BTFSC     VAL1,7
        GOTO      TT1
        BTFSC     VAL1,6
        GOTO      TT1
        BTFSC     VAL1,5
        GOTO      TT1
        BTFSC     VAL1,4
        GOTO      TT1
        BTFSC     VAL1,3
        GOTO      TT1
        BTFSC     VAL1,2
        GOTO      TT1
        BTFSC     VAL1,1
        GOTO      TT1
        BTFSC     VAL1,0
        GOTO      TT3

        GOTO      TT2

TT2     BTFSC     PORTA,2      ;WAS PORTA,1
        GOTO      KK1
        BTFSC     SPARE,0     ;BTFSC = BIT = 0 NEXT INSTRUCTION SKIPPED
        GOTO      KK1
        MOVLW     01h
        MOVWF     SPARE
        INCF      DIGIT3,1
        INCF      DIGIT4,1     ;WAS DIGIT4

KK1     BTFSS     PORTA,2      ;WAS PORTA,1
        GOTO      KK2
        BTFSS     SPARE,0
        GOTO      KK2
        CLRF      SPARE
        DECF      DIGIT3,1

KK2     GOTO      NEXT

TT3     BTFSC     PORTA,2
        GOTO      NEXT

```

```

MOVLW      1Fh
MOVWF      TERRY

MOVF        VAL4,W
SUBWF DIGIT4,W
MOVWF      VAL1
BTFSC      VAL1,7
GOTO       TT1
BTFSC      VAL1,6
GOTO       TT1
BTFSC      VAL1,5
GOTO       TT1
BTFSC      VAL1,4
GOTO       TT1
BTFSC      VAL1,3
GOTO       TT1
BTFSC      VAL1,2
GOTO       TT1
BTFSC      VAL1,1
GOTO       TT1
BTFSC      VAL1,0
GOTO       TT1

```

```

MOVLW      16h
MOVWF      TERRY
CLRF       VAL4

```

TT1

```

BTFSC      PORTA,2      ;IO IS NOW LOW SO SET UP DATA
*****PROGRAM MODE OFF*****
GOTO       NEXT

```

```

CLRF       VAL1

```

```

;WSB      BTFSS      PORTA,2
;          GOTO       WSB

```

```

CLRWDI
MOVLW      .20           ;we must now wait 4500 us
MOVWF      VAL3          WAS 10-25-23-45
PM1        MOVLW      .23;23      ;reusable values are val1 val5 val2 val3
MOVWF      VAL2

```

```

PM2        CLRWDI
DECFSZ     VAL2,1
GOTO       PM2
DECFSZ     VAL3,1
GOTO       PM1

```

```

MOVLW      .46;45
MOVLW      VAL3
B11        MOVLW      .46          ;reusable values are val1 val5 val3
MOVWF      VAL2
B12        CLRWDI
DECFSZ     VAL2,1

```

	GOTO	B12	
	DECFSZ	VAL3,1	
	GOTO	B11	
	BTFSC	PORTA,2	
	BSF	VAL1,0	
;	BCF	PORTA,0	
	MOVLW	.46	
	MOVWF	VAL3	
B21	MOVLW	.46	;reusable values are val1 val5 val2 val3
	MOVWF	VAL2	
B22	CLRWDT		
	DECFSZ	VAL2,1	
	GOTO	B22	
	DECFSZ	VAL3,1	
	GOTO	B21	
	BTFSC	PORTA,2	
	BSF	VAL1,1	
;	BSF	PORTA,0	
	MOVLW	.46	
	MOVWF	VAL3	
B31	MOVLW	.46	;reusable values are val1 val5 val2 val3
	MOVWF	VAL2	
B32	CLRWDT		
	DECFSZ	VAL2,1	
	GOTO	B32	
	DECFSZ	VAL3,1	
	GOTO	B31	
	BTFSC	PORTA,2	
	BSF	VAL1,2	
;	BCF	PORTA,0	
	MOVLW	.46	
	MOVWF	VAL3	
B41	MOVLW	.46	;reusable values are val1 val5 val2 val3
	MOVWF	VAL2	
B42	CLRWDT		
	DECFSZ	VAL2,1	
	GOTO	B42	
	DECFSZ	VAL3,1	
	GOTO	B41	
	BTFSC	PORTA,2	
	BFS	VAL1,3	
;	BFS	PORTA,0	
	MOVLW	.46	
	MOVWF	VAL3	
B51	MOVLW	.46	;reusable values are val1 val5 val2 val3
	MOVWF	VAL2	
B52	CLRWDT		
	DECFSZ	VAL2,1	
	GOTO	B52	
	DECFSZ	VAL3,1	
	GOTO	B51	
	BTFSC	PORTA,2	

	BSF	VAL1,4	
;	BCF	PORTA,0	
	MOVLW	.46	
	MOVWF	VAL3	
B61	MOVLW	.46	;reusable values are val1 val5 val3
	MOVWF	VAL2	
B62	CLRWDI		
	DECFSZ	VAL2,1	
	GOTO	B62	
	DECFSZ	VAL3,1	
	GOTO	B61	
	BTFSZ	PORTA,2	
	BSF	VAL1,5	
;	BSF	PORTA,0	
	MOVLW	.46	
	MOVWF	VAL3	
B71	MOVLW	.46	;reusable values are val1 val5 val2 val3
	MOVWF	VAL2	
B72	CLRWDI		
	DECFSZ	VAL2,1	
	GOTO	B72	
	DECFSZ	VAL3,1	
	GOTO	B71	
	BTFSZ	PORTA,2	
	BSF	VAL1,6	
;	BCF	PORTA,0	
	MOVLW	.46	
	MOVWF	VAL3	
B81	MOVLW	.46	;reusable values are val1 val5 val2 val3
	MOVWF	VAL2	
B82	CLRWDI		
	DECFSZ	VAL2,1	
	GOTO	B82	
	DECFSZ	VAL3,1	
	GOTO	B81	
	BTFSZ	PORTA,2	
	BSF	VAL1,7	
;	BCF	PORTA,0	
;	INCF	TERRY,1	
	BTFSZ	TERRY,4	
	GOTO	SSM	
	MOVLW	14h	
	MOVWF	TERRY	
SSM			
	MOVF	TERRY,0	
	MOVWF	FSR	
	MOVF	VAL1,0	
	MOVWF	INDIR	
	INCF	TERRY,1	
;	CLRF	DIGIT5	
;	CLRF	DIGIT6	

```

;      CLRF      DIGIT7
;      CLRF      DIGIT8

;      MOVF      VAL1,0
;      MOVWF     DIGIT4

;      MOVLW     .48
;      SUBWF     DIGIT5,W
;      MOVWF     VAL1
;      BTFSC     VAL1,0
;      GOTO      TX1
;      BTFSC     VAL1,1
;      GOTO      TX1
;      BTFSC     VAL1,2
;      GOTO      TX1
;      BTFSC     VAL1,3
;      GOTO      TX1
;      BTFSC     VAL1,4
;      GOTO      TX1
;      BTFSC     VAL1,5
;      GOTO      TX1
;      BTFSC     VAL1,6
;      GOTO      TX1
;      BTFSC     VAL1,7
;      GOTO      TX1

;      GOTO      TX2

;TX1
;      GOTO      NEXT
;*****END

;      SET_UP    ORG      200H

;      MOVF      DIGIT2,0      ;3,4,5
;      MOVWF     VAL5

;      GOTO      OUT3          ;NO PRIOR SETUP MODE      *****REMOVE *****

;      BTFSC     VAL5,7
;      GOTO      OUT3
;      BTFSC     VAL5,6
;      GOTO      OUT3
;      BTFSS     VAL5,5
;      GOTO      OUT3
;      BTFSS     VAL5,4
;      GOTO      OUT3

;      BTFSC     VAL5,3
;      GOTO      OUT1C
;      BTFSC     VAL5,2
;      GOTO      OUT1C
;      BTFSC     VAL5,1

```

GOTO	OUT1C
BTFSC	VAL5,0
GOTO	OUT1C
MOVLW	004FH ;was 004FH for no pull ups
OPTION	
GOTO	OUT4

OUT1C

BTFSC	VAL5,3
GOTO	OUT1B
BTFSC	VAL5,2
GOTO	OUT1B
BTFSC	VAL5,1
GOTO	OUT1B
BTFSS	VAL5,0
GOTO	OUT1B
MOVLW	004EH ;
OPTION	
GOTO	OUT4

OUT1B

BTFSC	VAL5,3
GOTO	OUT1A
BTFSC	VAL5,2
GOTO	OUT1A
BTFSS	VAL5,1
GOTO	OUT1A
BTFSC	VAL5,0
GOTO	OUT1A
MOVLW	004DH ;
OPTION	
GOTO	OUT4

OUT1A

BTFSC	VAL5,3
GOTO	OUT1
BTFSC	VAL5,2
GOTO	OUT1
BTFSS	VAL5,1
GOTO	OUT1
BTFSS	VAL5,0
GOTO	OUT1
MOVLW	004CH ;16
OPTION	
GOTO	OUT4

OUT1

BTFSC	VAL5,3
GOTO	OUT2
BTFSS	VAL5,2
GOTO	OUT2
BTFSC	VAL5,1
GOTO	OUT2
BTFSC	VAL5,0
GOTO	OUT2

```

        MOVLW      004BH ;16
        OPTION
        GOTO       OUT4

OUT2
        BTFSC      VAL5,3
        GOTO       OUT3
        BTFSS      VAL5,2
        GOTO       OUT3
        BTFSC      VAL5,1
        GOTO       OUT3
        BTFSS      VAL5,0
        GOTO       OUT3
        MOVLW      004AH ;32
        OPTION
        GOTO       OUT4

OUT3
        BTFSC      VAL5,3
        GOTO       OUT5
        BTFSS      VAL5,2
        GOTO       OUT5
        BTFSS      VAL5,1
        GOTO       OUT5
        BTFSC      VAL5,0
        GOTO       OUT5
        MOVLW      0049H      ;WAS 0049H
        OPTION
        GOTO       OUT4

OUT5
        ;RESET TAG
        MOVLW      004DH ;OE 2SEC OB = 13FAST 000DH FOR 1 PER SEC AND 000FH FOR SLOW
        OPTION
        MOVLW      .50      ;WAS 50 0A 2SEC OD = 13FAST 0BH FOR 1 PER SEC AND 09H FOR
SLOW
        MOVWF      DIGIT2 ;SPEED
        MOVLW      .50
        MOVWF      DIGIT3 ;SWITCH
        MOVLW      .00
        MOVWF      DIGIT4 ;COUNTER
        MOVLW      .00
        MOVWF      DIGIT5 ;AGE
        MOVLW      .00
        MOVWF      DIGIT6 ;
        MOVLW      .00
        MOVWF      DIGIT7 ;
        MOVLW      .00
        MOVWF      DIGIT8 ;
        MOVLW      .48
        MOVWF      DIGIT1 ;DATA 1
        MOVLW      .48
        MOVWF      DIGIT9 ;data 2
        MOVLW      .48
        MOVWF      DIGIT10
        MOVLW      .48
        MOVWF      DIGIT11

```

```
MOVLW      .48
MOVWF      DIGIT12
MOVLW      .48
MOVWF      DIGIT13
MOVLW      .48
MOVWF      DIGIT14
MOVLW      .65
MOVWF      DIGIT15
MOVLW      .80
MOVWF      DIGIT16
MOVLW      14h
MOVWF      TERRY
```

OUT4

```
INCFSZ DIGIT8,1
GOTO     OOTA
INCFSZ DIGIT7,1
GOTO     OOTA
INCFSZ DIGIT6,1
GOTO     OOTA
INCFSZ DIGIT5,1
GOTO     OOTA
```

OOTA

```
RETURN
```

CSUM

```
MOVF      DIGIT1,0
ADDWF     DIGIT2,0
ADDWF     DIGIT3,0
ADDWF     DIGIT4,0
ADDWF     DIGIT5,0
ADDWF     DIGIT6,0
ADDWF     DIGIT7,0
ADDWF     DIGIT8,0
ADDWF     DIGIT9,0
ADDWF     DIGIT10,0
ADDWF     DIGIT11,0
ADDWF     DIGIT12,0
ADDWF     DIGIT13,0
ADDWF     DIGIT14,0
ADDWF     DIGIT15,0
```

```
MOVWF     TERRY2
```

```
RETURN
```

```
end
```

CLAIMS:

1. An electronic tag which includes
 processor means programmed to provide a modulation control
 signal which includes unique identification data which at least identifies
5 the tag; and
 transmitter circuitry connected to the processor means and to an
 antenna for transmission of the unique identification data, the transmitter
 circuitry being powered by the modulation control signal.
2. An electronic tag as claimed in Claim 1, in which the
10 transmitter circuitry is exclusively powered by the modulation control
 signal of the processor means.
3. An electronic tag as claimed in Claim 1 or Claim 2, in which
 the transmitter circuitry includes passive components and a transistor
 defining oscillation circuitry directly driven by the processor means, the
15 transistor in combination with the passive components forming an
 integral part of the transmitter circuitry which is powered up by the
 modulation signal.
4. An electronic tag as claimed in any one of the preceding
 claims, in which the processor means is configured to provide the
20 modulation control signal with a first part followed by a second part, the
 first part including at least one high pulse of such a duration so as to
 provide sufficient power to the transmitter circuitry at least partially to
 stabilize it for transmission of the second part which includes data
 defined in a plurality of pulses which are of a substantially lesser
25 duration.

5. An electronic tag as claimed in Claim 4, in which the first part of the modulation control signal includes a plurality of high pulses that, in combination, provide an identification signal to a tag receiver for receiving a transmission from the electronic tag.
- 5 6. An electronic tag as claimed in Claim 4 or Claim 5, in which each pulse of the second part of the modulation signal includes a start portion for identifying a start of a bit and a data portion for identifying a state of the bit of data, the duration of the data portion selectively defining a high and a low state of the bit under control of the processor
10 means.
7. An electronic tag as claimed in Claim 6, in which the high state of the bit is defined by a shorter data portion during which the oscillator circuitry is switched off and the low state of the bit is defined by a longer data portion during which the oscillator circuitry is switched
15 off.
8. An electronic tag as claimed in any one of the preceding claims, in which the processor means is a micro-controller which includes an internal RC oscillator on which the modulation control signal is dependent and the micro-controller is arranged to enter a sleep mode
20 between data transmissions thereby to reduce power consumption.
9. An identification system which includes
a plurality of electronic tags, each tag including
processor means programmed to provide a modulation
control signal which includes unique identification data which at least
25 identifies the tag; and

transmitter circuitry connected to the processor means and to an antenna for transmission of the unique identification data, the transmitter circuitry being substantially powered by the modulation control signal; and

5 at least one electronic tag receiver configured to receive a transmission from the tag.

10. An identification system as claimed in Claim 9, in which the transmitter circuitry of the electronic tag is exclusively powered by the modulation control signal of the processor means.

10 11. An identification system as claimed in Claim 9 or Claim 10, in which the transmitter circuitry includes passive components and a transistor directly driven by the processor means, the transistor in combination with the passive components forming an integral part of the transmitter circuitry which is powered up by the modulation control
15 signal.

12. An identification system as claimed in any one of the preceding claims 9 to 11 inclusive, in which the processor means is configured to provide the modulation control signal with a first part followed by a second part, the first part including at least one high pulse
20 of such a duration so as to provide sufficient power to the transmitter circuitry at least partially to stabilize for transmission of the second part which includes data defined in a plurality of pulses which are of a substantially lesser duration.

13. An identification system as claimed in Claim 12, in which
25 the first part of the modulation control signal includes a plurality of high

pulses that, in combination, provide an identification signal to signal detection means of the electronic tag receiver for receiving a transmission from the electronic tag.

14. An identification system as claimed in Claim 12 or Claim 13,
5 in which each pulse of the second part of the modulation signal includes a start portion for identifying a start of a bit and a data portion for identifying a state of the bit of data, the duration of the data portion selectively defining a high and a low state of the bit under control of the processor means.

10 15. An identification system as claimed in Claim 14, in which the high bit is defined by a shorter data portion during which the transmitter circuitry is switched off and the low bit is defined by a longer data portion during which the transmitter circuitry is switched off.

15 16. A method of communicating data from an electronic tag, the method including driving transmitter circuitry of the tag with a modulation control signal which substantially powers the transmitter circuitry.

17. A method as claimed in Claim 16, in which the modulation control signal exclusively powers the transmitter circuitry.

20 18. A method as claimed in Claim 16 or Claim 17, which includes selectively modulating a fundamental frequency of an oscillator when data is transmitted and disabling the oscillator when data is not being transmitted.

19. A method as claimed in any one of the preceding claims 16 to 18 inclusive, in which the modulation control signal includes a first part followed by a second part, the first part including at least one high pulse of such a duration so as to provide sufficient power to the transmitter circuitry at least to partially stabilize it for transmission of the second part which includes data defined in a plurality of pulses which are of a substantially lesser duration.

20. A method as claimed in Claim 19, in which the first part of the modulation control signal includes a plurality of high pulses that, in combination, provide an identification signal to a tag receiver for receiving a transmission from the electronic tag.

21. A method as claimed in Claim 19 or Claim 20, in which each pulse of the second part of the modulation signal includes a start portion for identifying a start of a bit and a data portion for identifying a state of the bit of data, the duration of the data portion selectively defining a high and a low state of the bit under control of the processor means.

22. A method as claimed in Claim 21, in which the high bit is defined by a shorter data portion during which the modulation control signal is switched off and the low bit is defined by a longer data portion during which the modulation control signal is switched off.

23. A method as claimed in any one of the preceding claims 16 to 22 inclusive, in which the processor means is a micro-controller which includes an internal RC oscillator on which the modulation control signal

is dependent and the micro-controller is arranged to enter a sleep mode between data transmissions thereby to reduce power consumption.

24. A receiver for receiving a transmission from one of a plurality of electronic tags, the transmission including a first part and a second part and the receiver including
- 5 detection circuitry for detecting the first part and the second part of the transmission, the first part including at least one high pulse in response to which the receiver monitors reception of the second part which includes data defined in a plurality of pulses which are of a substantially lesser duration; and
- 10 timing means for timing the duration of each of the pulses in the second part and selectively generating a high or a low output defining a bit dependent upon the duration of the pulse.

25. A new electronic tag, substantially as herein described and
- 15 illustrated.

26. A new system, substantially as herein described and illustrated.

27. A new method of reducing power consumption in an electronic tag, substantially as herein described and illustrated.

28. A new receiver, substantially as herein described and
- 20 illustrated.