

## PIC Activity Board Description

Based around the PIC 16F877 micro-processor , the Activity board has a Microchip MCP2510 CAN device, 82C250 CAN driver and RS232 port interface. Programming or In Circuit Debugging is done through the Microchip ICD compatible RJ11-6 pin connector and the board has headers that allow expansion for the digital and analog parts not used by the Activity Board. For simple testing and evaluation a pushbutton switch and three coloured LEDs are also included. The CAN address for the board can be configured by putting up to 3 jumpers on a header which also includes a spot for a bus termination jumper.

### Power Supply:

Nine to twenty four volt power for the unit is supplied from either the Barrel 2.1mm (negative center) Power connector or the DB9 connector. Two zero ohm resistors are used to route the power so that power can be supplied exclusively from the DB-9 or supplied from either the DB-9 or the Barrel connector.

When both 0 Ohm resistors, R1 and R8, are not installed, the DB-9 pin 9 is completely isolated from the Activity board and the CAN bus pin 9 supply is neither used nor driven from the onboard power source.

With R1 and R8 place power comes from either the DB-9 or the Barrel Connector and will be sourced out onto the CAN bus power pin 9 when the Barrel connector is installed.

If only R8 is installed, power for the Activity Board is supplied from the Barrel Connector or the DB-9 CAN Connector but the Barrel Connector does not supply power to the Can Connector.

A 7805 regulator and filter capacitor create the Digital 5V supply for the microprocessor and CAN devices and if the 0 Ohm resistor R2 is installed, this 5V supply is also used for the A/D converter and routed to the Analogue expansion port. With R2 removed a 78L05 regulator can be installed dedicated to the analogue connector which also has a single point ground connection to the system ground to reduce interference.

### CAN Connector Connections:

DB-9P		
Description	PIN	Voltages
CAN_LO	2	2.5V to 1.0V
CAN_H I	7	2.5V to 4.0V
CAN_GND	3	0V
CAN_PWR	9	+8V to +24V

JP-3		
Description	Pins	Jumper Installed.
Termination	1-2	Terminates CAN bus at this node with 120 Ohm resistor.
Sel-0	3-4	0
Sel-1	5-6	0
Sel-2	7-8	0

**Micro-Processor System:**

The PIC 16F877 can receive it's clock from one of two locations: a Crystal X1 or the Clock Output from the on board MCP2510. To use X1 remove the R5 0 Ohm resistor and install two 18pF 0603 capacitors, C12 and C13, and a 20MHz crystal. If a reduced crystal drive is desired a resistor can replace the zero Ohm 0603 R??. When the MCP2510 oscillator is used, the MCP2510 must be programmed to run with a Divide by 1 setting in order to have the PIC oscillator run at 20Mhz. The processor can be reset with the Reset Pushbutton. The MCP2510 is reset on power up or with a software command.

The RS232 connection is done with a DB-9S connector that requires a straight through cable to connect to an IBM PC.

DB-9S	
Description	PIN
RS232-TX	2
RS232-RX	3
RS232-GND	5

## Serial Peripheral Interface (SPI™) bus:

The SPI bus interface uses the hardware Synchronous interface for clock and data lines. Device selection is accomplished with connections to Port D. The on board CAN MCP2510 device is hard wired as SPI\_SEL0 and the CANRF socket is hardwired as SPI\_SEL1; SPI\_SEL1 through to SPI\_SEL7 are brought out to pins 2,4,6,8,10,12 and 14 on Expansion Connector CONN1.

Also available on CONN1 are the PORTB pins including RB0/INT which is used by the MCP2510s. RB1 is connected to a Pushbutton and RB2, RB4 and RB5 are connected to GREEN, AMBER and RED LEDs respectively. The clock and compare counters, RC0 to RC2, are on pins 5, 3 and 1.

### CONN1 – Digital

RC2	1		2	SPI_SEL1 - RD1
RC1	3		4	SPI_SEL2 - RD2
RC0	5		6	SPI_SEL3 - RD3
RB5 - RED_LED	7		8	SPI_SEL4 - RD4
RB4 - AMBER_LED	9		10	SPI_SEL5 - RD5
RB2 - GREEN_LED	11		12	SPI_SEL6 - RD6
RB1 - BUTTON	13		14	SPI_SEL7 - RD7
RB0/INT	15		16	SPI_MISO
VCC	17		18	SPI_MOSI
VCC	19		20	SPI_CLK
VCC	21		22	GND
GND	23		24	GND

### CONN2 – Analogue

Board Revision 1.00 and 1.01

AVCC	1		2	AN4
RA4	3		4	AGND
AN3	5		6	AGND
AN2	7		8	AGND
AN1	9		10	AGND
AN0	11		12	AGND

Board Revision 1.02

AVCC	1		2	RA4
AN4	3		4	AGND
AN3	5		6	AGND
AN2	7		8	AGND
AN1	9		10	AGND
AN0	11		12	AGND

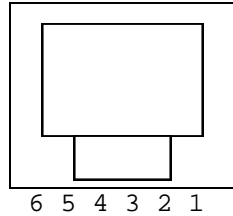
## Programming the PIC 16F877 Activity Board.

Port B pins RB3, RB6 and RB7 are reserved for the programming and In Circuit Debugging (ICD) interface and are therefore not brought out to the CONN1 digital connector. The pin connections for the RJ connector are the same as the Microchip PIC ICD programming module.

Pin Connections:

1. RB3
2. ISPC
3. ISPD
4. GND
5. VCC
6. Vpp

Pin #1 is the right hand pin on the connector from the circuit board edge side.



## PIC Activity Board Software Description:

Upon reset the Activity board firmware outputs an opening message containing the software version number and the node ID as read from the JP3 pins.

```
CANRF Activity Board Rev 0.18 Node #1
```

```
>
```

Commands can be entered from the serial port and the backspace key can be used to delete typing mistakes. Activity board commands all start with a letter and are followed by a number or another letter representing a sub command. Commands are terminated by a Carriage Return. Numbers are always in HEX so for example the number 10 is entered as an A.

### Node Addressing:

Location 0 of the EEROM holds the value that is added to the JP3 pins to define a unique node ID for CAN networking. A MAPCAN message with a node number equal to 0 is a global broadcast message. To avoid giving a node this value the sum of the EEROM location added to JP3 is incremented by 1 to generate a node ID from 1 to 255.

The serial communications port default Baud Rate is set to 115kbaud if location 5 has a value of 0. A non zero value will be written to the UART's SPBRG register and sets a different baud rate. To calculate the value for the SPBRG register, divide 12500 by the baud rate divided by 100 and store the value to EEROM location 5.

For example, to set the activity board communications port to 56K baud, divide  $12500/560 = 22$ . Don't forget to convert that to HEX 15.

```
>em5,15<CR>
```

```
>
```

Press the reset button to use the new baud rate. If the value installed into location 5 is wrong and the baud rate ends up as an indeterminate value the default 115kbps can be restored by holding down S1 while pressing reset.

### CAN Devices and Flags:

The flag bits in EEROM location 2 and 3 define other power up behaviors of the Activity board and are copied from EEROM to a RAM image after a reset and so changes do not take effect till the next reset or power up.

There is an onboard MCP2510 device located near the male DB9 connector. This is CAN device 0 and is used as the active CAN device when a 0 is programmed into EEROM location 1. The CANRF module is used when a 1 is programmed into EEROM location 1.

To enable the on board CAN connector.

```
>em1,0<CR>
```

```
>
```

To enable the CANRF module.

```
>em1,1<CR>
```

```
>
```

Then press reset.

The CAN bit rate is 20kbps by default specifically for the 916MHz CANRF. An alternate default bit rate, like 10kbps for the 433MHz CANRF, can be configured by setting Bit 4 in EEROM location 2. When this bit is set, the startup code uses EEROM locations 10, 11 and 12 for the three MCP2510 bit rate configuration registers.

The values 0x31, 0xBF, 0x02 are used to set 10kbps when the MCP2510 crystal is 20MHz.

To enable EEROM configuration Bytes

```
>em2, 10<CR>
```

```
>
```

To enable EEROM CAN Configuration Bytes and the RX and TX LEDs.

```
>em2, 15<CR>
```

```
>
```

### **LED Indicators:**

There are three LED indicators located near the wire wrap area beside switch S1. The LEDs are marked RB2, RB4 and RB5 which indicate which processor pin is used to drive the respective LED. The RED LED (RB5) can be configured to blink every time a CAN message is received while the GREEN LED can be set to blink every time the board transmits a CAN packet.

The YELLOW LED has two possible functions. It can be enabled to blink in time with the heartbeat messages and when disabled will turn on whenever a Switch Down Message is received and turn off when the Switch Up message is received.

The LEDs are enabled by the first 3 bits of the EEROM location 2:

Bit 0 – Show RX Message (RED LED)

Bit 1 – Show TX Message (GREEN LED)

Bit 2 – Show Heartbeat (YELLOW LED)

**HeartBeats:**

When EEROM location 2 Bit3 is set.

- Transmit CAN Heartbeat ALIVE message.

The heartbeat clock period is fixed at 1 per second. If the Heartbeat YELLOW LED is enabled the LED changes from a short blink to a 1 second on and one second off display.

The heartbeat timer can also be used to trigger a series of node reports. If Bit 4 of EEROM location 3 is set, the node is initialized as the master. It counts a number of heartbeats based on the value stored in EEROM location 4 and then issues a MAPCAN\_SYNC\_CLOCK message. This message contains the master real Time of Day Clock and is used to synchronize all the other nodes to the same Time of Day.

A node which receives this message responds with its image of all the heartbeats that were received since the previous sync message. This is a bit map allowing up to 47 nodes contained in the last 6 bytes of the MAPCAN\_CMD\_MAP message. Remember, no more than 47 nodes are allowed so don't set the NodeID in EEROM location 0 to a value greater than 39 if you want to use this.

e.g. Below is a heartbeat from node #1, then the master SYNC\_CLOCK message from node #6 and the response from node #1 with the MAP of visible nodes. In this case, bits 5 and 6 are set in data byte 2 which means nodes 5,6 are visible to node 1.

```
00:05:59.41 ID:101 DLC:2 00 10
00:06:00.45 ID:106 DLC:8 00 11 00 F7 00 06 00 2C
00:06:00.45 ID:001 DLC:8 00 12 60 00 00 00 00 00
```

When the master sends out the SYNC\_CLOCK message all other nodes respond with their MAP message so there will be a flurry of back to back messages after each SYNC\_CLOCK. The MAP messages are important because they show which nodes have access to each other. On a wired physical network, the MAPs will all match because the wired network can receive all messages from all nodes. The Wireless CANRF network can suffer from interference and not all nodes may be visible to each other and the MAP shows the interactions between nodes. The master could (but doesn't), analyze these maps and determine which nodes have not received the SYNC\_CLOCK message. It could then ask another node to also issue a SYNC\_CLOCK so that all nodes would remain synchronized.

The request to have another node issue the SYNC\_CLOCK message is called a SET\_HIS\_CLOCK command and includes, as an argument, the ID of the node who's clock must be set. On reception of this message the node issues a SET\_CLOCK command to the targeted node which responds with a GOT\_CLOCK message.

e.g. Node 9 asks Node 1 to set the clock of Node 6.

```
>ct 109 01 15 06
Sending:109 DLC=3 Data:01 15 06
>
20:45:40.21 ID:101 DLC:8 06 14 0A 65 14 2D 28 13
20:45:40.22 ID:106 DLC:8 01 16 0A 65 14 2D 28 13
```

To query a node's Time of Day clock you can send a TELL\_CLOCK to a node. The node responds with a MY\_CLOCK message that includes the Time of Day of that node.

e.g. Node 9 wishes to know the Time of Day clock value of Node 6.

```
>ct 109 06 17
Sending:109 DLC=2 Data:06 17
>
20:39:30.28 ID:106 DLC:8 09 18 0A 65 14 27 1E 1C
```

## I/O Devices:

The Activity board has firmware demonstrating two different MAPCAN I/O devices. The LEDs are a USER\_DEFINED device class and the pushbutton has been defined as a KEYPAD device. Earlier firmware uses the NodeID value (which should therefore be restricted to a value of 1..15) as the Device ID within the Device class. Later versions have two EEROM locations so the ButtonID and the LampID, (EEROM locations 40 and 45 respectively), can be set with a unique value.

Message to the LEDs are Device messages where the first two bits of the ID are 01 and the Client/Server bit is set. As always, a Client task sends a command to the Server device so the NodeID of the task makes up the rest of the ID.

The response from the Device also has the first two bits configured as a Device 01 and now the Client/Server Bit is cleared and the ID portion consist of the Device Class and the Device ID.



## The LAMP Device:

e.g. To make the YELLOW LED (LED 1) blink with a period of 0.5 seconds on and 0.5 seconds off send the following CAN message to Lamp Device # 1 from Task 0x20. In order to control this LED the Heartbeat Flag in EEROM location 2 must not be set.

```
>ct 320 31 3e 1 2 5 5
Sending:320 DLC=6 Data:31 62 01 02 05 05
```

A Device message from Task 0x20 to a device has 6 bytes.  
 The destination device is Class 3, Number 1 represented as 0x31.  
 This is a DEV\_CMD represented by the number 0x62  
 For LED 1 (The yellow one)  
 And we want the LED to flash (Lamp state 2)  
 On for 5 \* 0.1 Seconds and Off for 5 \* 0.1 Seconds.

If the device correctly receives the message it returns the following response echoing the above command.

```
>
00:56:24.56 ID:231 DLC:4 20 62 01 02
```

A message FROM device Class 3 Number 1 to Node 20  
 Device 0x31 received a DEV\_CMD (0x62)  
 For LED 1 (The yellow one)  
 To Flash at a rate passed in the DEV\_CMD message.

Note that the Destination of the message is a Task or Thread and not a Device because devices are not allowed to talk to each other directly.

## The Button Device :

The user pushbutton software is configured as a KEYPAD Device class and is configured so that each press of the switch creates a switch down message and, if the switch is held down, a continuous stream of switch down messages. The delay between the first message and the rate are programmable as part of the device configuration. When the switch is released a switch up message is sent.

The Keypad Device also responds to requests for current switch state. Here the KEYPAD Class is 0x10 and the keypad Device is #1 so Task #0x20 sends a Client Keypad Status request (0x04) to Device 0x11 and gets a response back from the device with the keypad value. A larger keypad may have more data bytes.

```
>ct 320 11 4
Sending:320 DLC=2 Data:11 04
>
01:16:07.08 ID:211 DLC:3 20 04 00
```

## PIC Activity Board Command set:

(lower case characters)

### [c] CAN Commands

Each command starts with a lowercase 'c' immediately followed by one of the sub-commands.

**@** - Reset CAN  
Sends reset to current CAN device and puts it into 'config' mode .

**Bx,y,z** - BitRate  
Sets CAN bit rate.  
Only works if the MPC2510 is in CONFIG mode.  
e.g. CG4<CR> CB1<CR> CG0<CR>

With no arguments bit rate is set to 20kbps.

If there is one argument the value of the argument is a table index from 1..9  
An index value of 0 means that EEROM locations 8, 9 and 10 are used for bitrates.

The table, with values in the order CTRL1, CTRL1, CTRL2 is:

```
// Following clock defines are for a 20MHz Crystal on MCP2510.
// 800kBPS and 1000kBPS do not meet minimum 26 clock bus
// idle problem on Rev A Silicon Errata Sheet.
const BYTE DefaultCANSpeed[27] = {
    0x31,0xBF,0x02,    // 10kBPS
    0x18,0xBD,0x02,    // 20kBPS
    0x09,0xBF,0x02,    // 50kBPS
    0x04,0xBF,0x02,    // 100kBPS
    0x03,0xBF,0x02,    // 125kBPS
    0x01,0xBF,0x02,    // 250kBPS
    0x00,0xBF,0x02,    // 500kBPS
    0x00,0xDF,0x02,    // 800kBPS
    0x00,0xA1,0x01     // 1000kBPS
};
```

If all three arguments are supplied they are sent to the MCP2510 CNTRLx registers and are used in the order: x=CTRL0, y=CTRL1, z=CTRL2.

Bit rate defaults to 20kbps with 75% sample point.

### C - Interrupt Flags Value

The Interrupt routine clears the Interrupt enable for the interrupt source and then sets a bit in the Interrupt flag register. The mainline code checks this flag and determines what it must do and if interrupts will be re-enabled.

```
RX0IF    --    MsgStatus.0 = 1; RxBuffer 0 has new data.
RX1IF    --    MsgStatus.1 = 1; RxBuffer 1 has new data
TX0IF    --    MsgStatus.2 = 1; TxBuffer 0 is empty
TX1IF    --    MsgStatus.3 = 1; TxBuffer 1 is empty
TX2IF    --    MsgStatus.4 = 1; TxBuffer 2 is empty
ERRIF    --    MsgStatus.5 = 1; Error Has ocured
WAKIF    --    MsgStatus.6 = 1; Device has woken.
```

- D** - Dump CAN Registers  
Dump the MCP2510 registers in the same format as the table in the data sheet.
- E** - Error Flags  
Mnemonic description of Error Flags.  
Text printed when specific error bit is active.
- EWARN*  
*RXWAR*  
*TXWAR*  
*RXEP*  
*TXEP*  
*TXBO*  
*RX0OVR*  
*RX1OVR*
- Gn** - Go to n Mode  
bSet MCP2510 into Normal(0), Sleep(1), loop(2), Listen(3) or Config(4) mode.  
---- Note that as per Errata, node must first go into loopback before listen mode.
- '?'** - Display Signon message.
- H** - Toggle Heartbeat Flag (Not saved to EEROM)
- I** - INT status  
Mnemonic description of Interrupt flags.
- |        |    |                          |
|--------|----|--------------------------|
| RX0IF  | -- | RxBuffer 0 has new data. |
| RX1IF  | -- | RxBuffer 1 has new data  |
| TX0IF  | -- | TxBuffer 0 is empty      |
| TX1IF  | -- | TxBuffer 1 is empty      |
| TX2IF  | -- | TxBuffer 2 is empty      |
| ERRIF  | -- | Error has occurred       |
| WAKIF  | -- | Device has woken.        |
| MERRIF | -- | Multiple errors exist    |
- M** - Report MICKEY and ALIVE MAP.
- Rx** - Read CAN Register  
Read CAN register from hex address x.
- S** - CAN Status Bytes  
Display the important CAN status bytes:
- ```

STAT:      MCP_status();
CANSTAT:   MCP_CANStatus();
INTSTAT:   MCP_IntStatus();
           MCP_WRITE(EFLG,1);           // Clear error flag
           MsgBuf[0] = DecArg & 0x1F;  // Clear error flags.
           MCP_WRITE(CANINTF,1);
EFLG:      MCP_READ(EFLG,1);
TEC:       MCP_READ(TEC,2);
REC:
CNF3:      MCP_READ(CNF3,3);

```
- T{X}** - Transmit CAN message  
Transmit a standard or extended message in the format:  
T iii {dd}

or  
TX eeeee {dd}

**U** - Update the Time of Day clocks and issue a SET\_CLOCK MAPCAN message.

**Wx,y** - Write CAN register  
Write the hex value y to hex address x.

End of CAN commands.

- [e]** -- EEROM dump and modify commands.  
The address is decimal and the argument is hex.
- d** - edn -- Dump EEROM page n x [0..1] to screen.
  - f** - efx -- Fill EEROM with hex value x
  - m** - emn,x -- Modify memory location n with value hex value x.
  - p** - epn -- Put EEROM location n to display
- [v]** -- Version Command.

**EEROM parameter description Rev 0.17:**

| Address | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 00      | – Node ID Added to 3 bit value read from Address Selection header.                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 01      | – SPI bus CAN Device Selection ID. <ul style="list-style-type: none"><li>a. 0 for Onboard MCP2510</li><li>b. 1 for CAN RF MCP2510</li></ul>                                                                                                                                                                                                                                                                                                                                                                                   |
| 02      | – CAN flags for setting mode and LED status. <ul style="list-style-type: none"><li>a. Bit 0 – Flash RED LED when a message is received.</li><li>b. Bit 1 – Flash GREEN LED when a message is transmitted.</li><li>c. Bit 2 – Flash YELLOW LED for heartbeat indication.</li><li>d. Bit 3 – Send a CAN message once per Heartbeat</li><li>e. Bit 4 – Get Bit rate from EEROM location 10</li><li>f. Bit 5 – Use CAN Data at location 20 as Heartbeat message.</li><li>g. Bit 6 – Show Bus Status Interrupt messages.</li></ul> |
| 03      | – System Flags for program operations. <ul style="list-style-type: none"><li>a. Bit 0..3 – Leave set at 0</li><li>b. Bit 4 – Set to transmit SYNC CLOCK message.</li><li>c. Bit 5..7 – Leave set at 0</li></ul>                                                                                                                                                                                                                                                                                                               |
| 04      | – Number of seconds between SYNC CLOCK messages.                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 10      | – CNF1 Default value when B command index is 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 11      | – CNF2 Default value when B command index is 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 12      | – CNF3 Default value when B command index is 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 20      | – 2 x Period of CAN Heartbeat in 10Ms Ticks.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 21      | – High Byte of 11 bit ID or 29 bit ID                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 22      | – Low Byte of 11 bit ID or Middle High of 29 bit ID                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 23      | – Middle Low Byte of 29 bit ID                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 24      | – Low Byte of 29 bit ID                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 25      | – Flag for 11 bit (0) or 29 bit (1) ID                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 26      | – DLC of message                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 27      | to 34 are the 8 data bytes.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

**a. EEROM parameter description Rev 0.18 and above:**

| Address | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 00      | – Node ID Added to 3 bit value read from Address Selection header.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 01      | – SPI bus CAN Device Selection ID. <ul style="list-style-type: none"> <li>a. 0 for Onboard MCP2510</li> <li>b. 1 for CAN RF MCP2510</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 02      | – CAN flags for setting mode and LED status. <ul style="list-style-type: none"> <li>a. Bit 0 – Flash RED LED when a message is received.</li> <li>a. Bit 1 – Flash GREEN LED when a message is transmitted.</li> <li>b. Bit 2 – Flash YELLOW LED for heartbeat indication.</li> <li>c. Bit 3 – Send a CAN message once per Heartbeat</li> <li>d. Bit 4 – Get CAN Bit rate from EEROM location 10</li> <li>e. Bit 5 – Use CAN Data at location 20 as Heartbeat message.</li> <li>f. Bit 6 – Show Bus Status Interrupt messages.</li> <li>g. Bit 7 – Show A/D AN0 at end of each received message. (RSSI).</li> </ul> |
| 03      | – System Flags <ul style="list-style-type: none"> <li>a. Bit 0 – Master Flag – Set if this modules sends out SYNC_CLOCK.</li> <li>b. Bit 1 – add Packet count to last byte of message if room exists.</li> <li>c. Bit 4 – Set to transmit SYNC CLOCK message.</li> <li>d. Bit 5..7 – Leave set at 0</li> </ul>                                                                                                                                                                                                                                                                                                      |
| 04      | – Number of seconds between SYNC CLOCK messages.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 05      | – Serial Port Baud Rate. Default 115kBaud if equal to 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 06      | – Comm Flags <ul style="list-style-type: none"> <li>a. Bit 0 – For local Echo and backspace edit enable.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 10      | – CNF1 Default value when B command index is 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 11      | – CNF2 Default value when B command index is 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 12      | – CNF3 Default value when B command index is 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 20      | – 2 x Period of CAN Heartbeat in 10Ms Ticks.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 21      | – High Byte of 11 bit ID or 29 bit ID                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 22      | – Low Byte of 11 bit ID or Middle High of 29 bit ID                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 23      | – Middle Low Byte of 29 bit ID                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 24      | – Low Byte of 29 bit ID                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 25      | – Flag for 11 bit (0) or 29 bit (1) ID                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 26      | – DLC of message                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 27      | – 34 are the 8 data bytes.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 40      | – Button Device ID (1..15)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 41      | – Button Debounce time * 10ms                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 42      | – Delay before Repeat                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 43      | – Delay between Repeat Button Down messages.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 44      | – Lamp Device ID (1..15)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |