**Cerebot32MX4 Bluetooth Remote Control**

**Overview**

This project demonstrates the use of the pmodlib C library in conjunction with the PmodJSTK, PmodCLS, Pmod8LD, and PmodBT2 and Cerebot32MX4 microcontroller as a Bluetooth remote control for a robot. The companion project (the robot) is the Cerebot32MX4 Bluetooth Robot.

**Functional Description**

This project consists of a custom made remote chassis that will house the Cerebot32MX4 along with two PmodJSTK, battery case, PmodCLS, PmodBT2, and Pmod8LD. Remote input is taken in through two PmodJSTK modules, each monitoring a separate axis. JSTK1: y-axis (fwd/rev) JSTK2: x-axis(left/right). Commands sent to the robot are transmitted using the PmodBT2 Bluetooth module. Information received from the robot (rpm, battery voltage and direction) are displayed on the PmodCLS. Joystick calibration instructions and initialization status are also displayed on the PmodCLS during startup.

Once power is applied to the remote, the Bluetooth module initialization process begins, "Initializing" is displayed on the PmodCLS until complete. Once the PmodBT2 has been initialized joystick calibration begins, the goal of calibration is to determine the minimum and maximum bounds for the X and Y axes. PmodJSTK LD1 will illuminate on the joystick currently in calibration, the PmodCLS will display the direction and axis to calibrate. To calibrate, apply full deflection in the direction requested (x-, x+, y-,y+) on the PmodJSTK with LD1 illuminated. While maintaining full deflection , press BTN1 on the same PmodJSTK. Repeat as directed for other axes. If either axis range calculated at the end of calibration is inadequate the process will repeat.

After calibration, the remote will attempt to locate the address of the robot by performing a Bluetooth inquiry for "CEREBOTROBOT"(configurable), resolving this name to an address. The remote will then attempt to connect, "Connecting..." will be displayed on the PmodCLS during the inquiry and connection process. Once connected, the Pmod8LD will illuminate, Left/Right wheel RPM, robot battery voltage and direction of travel will be displayed on the PmodCLS. If the remote fails to connect, the inquiry and connect process will execute a specified number of times (configurable), once the number of tries expires the remote will require a reset. If the remote loses its connection to the robot during operation, it will initiate the inquiry process and attempt to reconnect.

Once the remote has successfully connected , deflecting the fwd/rev joystick in the y+ direction will move the robot forward, y- will cause it to reverse direction. Deflecting the left/right joystick in the x- direction turn the robot left, x+ will turn the robot right, x axis movements only turn the robot while it is motion



Joystick movements will transmit the percentage of deflection on the axis monitored as displayed in the figure to the left. Position of the joystick in the positive or negative position relative to its axis is determined and sent to the robot indicating FWD/REV or LEFT/RIGHT. Due to potential inaccuracies during joystick calibration, axis limits and range values are recalculated during operation when a min or max range value has been exceeded.

Bluetooth connection state is monitored by polling the STATUS pin of the PmodBT2, which is driven high when connected and low otherwise.

Source code for the remote is located in the directory titled "Cerebot Remote". This project was created using MPLAB IDE v8.80 and written entirely in C utilizing the Microchip and pmodlib libraries.

**Configuration**\*All settings referenced in this section are in "cerebot\_robot\_remote.c"

Processor: configuration settings are located in the "PIC32 Configuration Settings" section.

Pmod IO/Ports - IO ports/bits and channels have been defined as macros for easy configuration under //IO PORT/CHANNEL DEFINITIONS. Changes in hardware location must be reflected in this section.

Bluetooth Communication/Status - The Bluetooth name of the remote control is a macro defined as BLUETOOTH\_REMOTE\_NAME, the remote side (the robot) is defined as BLUETOOTH\_ROBOT\_NAME and is used during inquiry to resolve the address of the robot. The use of the Pmod8LD for connection status is optional, to use LD1 on the Cerebot 32MX4 modify macro BLUETOOTH\_STAT\_LED, a value of 0 will use LD1, 1 will use PmodLD8, any other value will disable connection status notification. Additional Bluetooth parameters are set in initPmodBT2(), detailed explanations of these settings are available in the Roving Networks RN-42 Advanced User Manual.

**Clock Rates:**

Processor Freq: 80Mhz

Peripheral Bus: 40Mhz

**Port Bitrates:**

UART1 - PmodCLS: 9600

UART2 - PmodBT2: 115200

SPI1 - PmodJSTK: 156250

SPI2 - PmodJSTK: 156250

**Communication Protocol :**

Messages sent/received via serial communication through the PmodBT2 utilize structs CEREBOT\_REMOTE\_MSG and CEREBOT\_ROBOT\_MSG located in "cerebot\_robot\_remote\_types.h" .

**Remote State Diagram:**

The following state transition diagram represents the main program loop of the remote keying on specific events that trigger a transition from one state to the next. State names line up with states present in appTask() prefixed with STATE\_.



**Assembly:**

The photos below show how the remote should look after assembly. Tables detailing the electrical connections of the different components are provided below the photos.

|  |
| --- |
|  |
| C:\Users\ryan\Desktop\BluetoothRobotDocs\ControllerRear.bmp |

The table below details the electrical connections between the different components of the remote. All jumpers are in their default positions except J12 which is shorted on "External Power".

|  |  |  |
| --- | --- | --- |
| Cerebot 32MX4 | | |
| Header | Description | Connections |
| J1 | Left/Right Joystick (SPI1) | |  |  |  | | --- | --- | --- | | Pin | Peripheral Header | Peripheral Pin | | SS | PmodJSTK - J1 | 1 | | SDO | PmodJSTK - J1 | 2 | | SDI | PmodJSTK - J1 | 3 | | SCK | PmodJSTK - J1 | 4 | | GND | PmodJSTK - J1 | 5 | | VCC | PmodJSTK - J1 | 6 | |
| JB | Fwd/Rev Joystick(SPI2) | |  |  |  | | --- | --- | --- | | Pin | Peripheral Header | Peripheral Pin | | 1 | PmodJSTK - J1 | 1 | | 2 | PmodJSTK - J1 | 2 | | 3 | PmodJSTK - J1 | 3 | | 4 | PmodJSTK - J1 | 4 | | 5 | PmodJSTK - J1 | 5 | | 6 | PmodJSTK - J1 | 6 | |
| JC | Bluetooth connected status indicator | |  |  |  | | --- | --- | --- | | Pin | Peripheral Header | Peripheral Pin | | 1 | Pmod8LD - J1 | 1 | | 2 | Pmod8LD - J1 | 2 | | 3 | Pmod8LD - J1 | 3 | | 4 | Pmod8LD - J1 | 4 | | 5 | Pmod8LD - J1 | 5 | | 6 | Pmod8LD - J1 | 6 | | 7 | Pmod8LD - J1 | 7 | | 8 | Pmod8LD - J1 | 8 | | 9 | Pmod8LD - J1 | 9 | | 10 | Pmod8LD - J1 | 10 | | 11 | Pmod8LD - J1 | 11 | | 12 | Pmod8LD - J1 | 12 | |
| JE | LCD status display | |  |  |  | | --- | --- | --- | | Pin | Peripheral Header | Peripheral Pin | | 1 | PmodCLS - J2 | 1 | | 2 | PmodCLS - J2 | 2 | | 3 | PmodCLS - J2 | 3 | | 4 | PmodCLS - J2 | 4 | | 5 | PmodCLS - J2 | 5 | | 6 | PmodCLS - J2 | 6 | |
| JH | Bluetooth UART connection (Using UART X-Over Cable) | |  |  |  | | --- | --- | --- | | Pin | Peripheral Header | Peripheral Pin | | 1 | PmodBT2 - J1 | 1 | | 2 | PmodBT2 - J1 | 4 | | 3 | PmodBT2 - J1 | 3 | | 4 | PmodBT2 - J1 | 2 | | 5 | PmodBT2 - J1 | 5 | | 6 | PmodBT2 - J1 | 6 | |

|  |  |  |
| --- | --- | --- |
| Cerebot 32MX4 (Continued) | | |
| Header | Description | Connections |
| JJ | Bluetooth Connection and Status | |  |  |  | | --- | --- | --- | | Pin | Peripheral Header | Peripheral Pin | | 1 | PmodBT2 - J1 | 7 | | 2 | PmodBT2 - J1 | 8 | | 3 | PmodBT2 - J1 | 9 | | 4 | PmodBT2 - J1 | 10 | | 5 | PmodBT2 - J1 | 11 | | 6 | PmodBT2 - J1 | 12 | |
| J14 | External Power (Battery Holder) | |  |  |  | | --- | --- | --- | | Pin | Peripheral Header | Peripheral Pin | | GND | Battery | Black | | VEXT | Battery | Red | |

|  |  |  |
| --- | --- | --- |
| PmodCLS | | |
| Header | Description | Connections |
| J2 | UART Communication | |  |  |  | | --- | --- | --- | | Pin | Peripheral Header | Peripheral Pin | | 1 | Cerebot 32MX4 - JE | 1 | | 2 | Cerebot 32MX4 - JE | 2 | | 3 | Cerebot 32MX4 - JE | 3 | | 4 | Cerebot 32MX4 - JE | 4 | | 5 | Cerebot 32MX4 - JE | 5 | | 6 | Cerebot 32MX4 - JE | 6 | |
| JP2 | Mode Selection | REV E REV D MD0 = shorted MD0 = open  MD1 = open MD1 = shorted  MD2 = shorted MD2 = open |

|  |  |  |
| --- | --- | --- |
| PmodJSTK(Left) | | |
| Header | Description | Connections |
| J1 | SPI Communication | |  |  |  | | --- | --- | --- | | Pin | Peripheral Header | Peripheral Pin | | 1 | Cerebot 32MX4 - JB | 1 | | 2 | Cerebot 32MX4 - JB | 2 | | 3 | Cerebot 32MX4 - JB | 3 | | 4 | Cerebot 32MX4 - JB | 4 | | 5 | Cerebot 32MX4 - JB | 5 | | 6 | Cerebot 32MX4 - JB | 6 | |

|  |  |  |
| --- | --- | --- |
| PmodJSTK(Right) | | |
| Header | Description | Connections |
| J1 | SPI Communication | |  |  |  | | --- | --- | --- | | Pin | Peripheral Header | Peripheral Pin | | 1 | Cerebot 32MX4 - J1 | SS | | 2 | Cerebot 32MX4 - J1 | SDO | | 3 | Cerebot 32MX4 - J1 | SDI | | 4 | Cerebot 32MX4 - J1 | SCK | | 5 | Cerebot 32MX4 - J1 | GND | | 6 | Cerebot 32MX4 - J1 | VCC | |

|  |  |  |
| --- | --- | --- |
| PmodBT2 | | |
| Header | Description | Connections |
| J1 | Bluetooth UART connection (Using UART X-Over Cable)/ Bluetooth Connection and Status | |  |  |  | | --- | --- | --- | | Pin | Peripheral Header | Peripheral Pin | | 1 | Cerebot 32MX4 - JH | 1 | | 2 | Cerebot 32MX4 - JH | 4 | | 3 | Cerebot 32MX4 - JH | 3 | | 4 | Cerebot 32MX4 - JH | 2 | | 5 | Cerebot 32MX4 - JH | 5 | | 6 | Cerebot 32MX4 - JH | 6 | | 7 | Cerebot 32MX4 - JJ | 1 | | 8 | Cerebot 32MX4 - JJ | 2 | | 9 | Cerebot 32MX4 - JJ | 3 | | 10 | Cerebot 32MX4 - JJ | 4 | | 11 | Cerebot 32MX4 - JJ | 5 | | 12 | Cerebot 32MX4 - JJ | 6 | |
| JP1 - 4 | See PmodBT2 RM | Open |

**Parts List**

**Digilent**

(1) Cerebot 32MX4

(1) PmodBT2

(2) PmodJSTK

(1) Pmod8LD

(1) UART Crossover Cable

(5) 6 Pin Cable Connector

(1) 2x6 Pin Cable

(4) 6 Pin Header & Gender Changer

(1) 2x6 Pin Header

(4) Standoffs (contains 4ea)

(1) Battery Holder (4 x AA)

(1) Base Plate Expansion Kit

(1) Angled Plate Expansion Kit

(1) Angled Plate Expansion (90̊ flange)

(2) Pmod Clip