# Temperature-Based Remote Motor Control via the CAN Network

## Objective

Use the Melexis IR sensor to control the speed of a motor within configurable set points.

## Resources

FreeRTOS, SMBus IR sensor, CAN controllers, Input Capture module, Output Compare module

## Description

Figure 1 gives an overall view of the project components. Please note that rather than two separate Cerebot boards we will use two CAN controllers in a single PIC32. (i.e., not loopback mode) The CAN controller on the left will be the “control” unit (CAN1, J9) and the one on the right is the “IO” unit (CAN2, J10). Their operation is as follows.

### IO Unit – implemented using one or more tasks

* The IO unit will read the temperature and the motor speed every 500 ms.
* The unit will receive Remote Frames (Remote Transmission Request) from the Control unit and respond with the temperature (F), average motor speed (RPS), and PWM setting (%).
* The unit will also receive Data Frames with desired PWM settings for the motor.
* The unit powers up with a PWM value of zero.

### Control Unit – implemented using one or more tasks

* The Control unit will have two modes: configuration and operational
* BTN1, operated as a push on/push off and controlling LED1, will cause the unit to toggle between the two modes. (LED1 “on” indicates “Operational” mode.)
* The unit powers up in Configuration Mode with a blank LCD and no set points
* The unit sends Remote Frames to the IO unit every two seconds.

#### Configuration Mode

* Displays current temperature (F) in center of LCD Line 1
* BTN3 establishes the LO set point; BTN2 the HI set point
* Once established, the set points are displayed at the outside edges of Line 2 with LO on the left
* If the HI set point is less than the LO set points, then they are cleared and the bottom line cleared
* BTN1 causes a change to Operational Mode.

#### Operational Mode

* Updates the LCD when a sensor message is received, using the most recent data.
* LCD displays PWM and RPS on the outer edges of Line1, and LO, TEMP, and HI on Line 2 (see Figure 2)
* Responds to the sensor message with a PWM message.
* If TEMP < LO, PWM=20%. If TEMP > HI, PWM=95%. In between is linear function over [30-85]% (see Figure 3)
* BTN1 causes a change to Configuration Mode, but does not alter the set points

## Instrumentation (LED Behavior)

* LED1 indicates the state of the Control Unit (ON = Operational)
* LEDA toggles when the Control Unit requests a sensor message
* LEDB toggles when the Control Unit receives a sensor message
* LEDC toggles when the Control Unit sends a PWM message
* LEDD toggles when the IO Unit applies the received PWM setting

## Equipment Connections

* IR Sensor – Attach to J7 (I2C1) such that the green wire is attached to SCL1 and the red wire is attached to 3V3. (See p. 17 of the Cerebot RM.) ***Do not write to the IR Sensor!!!*** Instead, follow Figure 8 from the IR datasheet to read the temperature, and then follow p. 27 to determine the object (TOBJ1) temperature in degrees Fahrenheit.
* CAN Cables – Connect J9 to J10. Make sure that “like” is connected to “like”, i.e., CANH to CANH and 3V3 to 3V3. You also need to move JP1 and JP2 into the CAN position. Lastly, position JP5 or JP7 to provide a terminating resistor. (See p. 14 of the Cerebot RM.)
* Linear Motor – No changes needed to the ECE 341 platform.

Note: The signals on J7, J9, and J10 are replicated on both rows.

## Figures

|  |
| --- |
| IMG_0001.tif |
| Figure 1 – System Structure |

|  |
| --- |
| IMG.jpg |
| Figure 2 – LCD Format |
| Legend: LO/HI Set points – L/H; Object Temperature (F) – O; PWM – X; Speed (RPS) – y. |

|  |
| --- |
| IMG_0002.tif |
| Figure 3 – PWM versus Temperature Set Points |