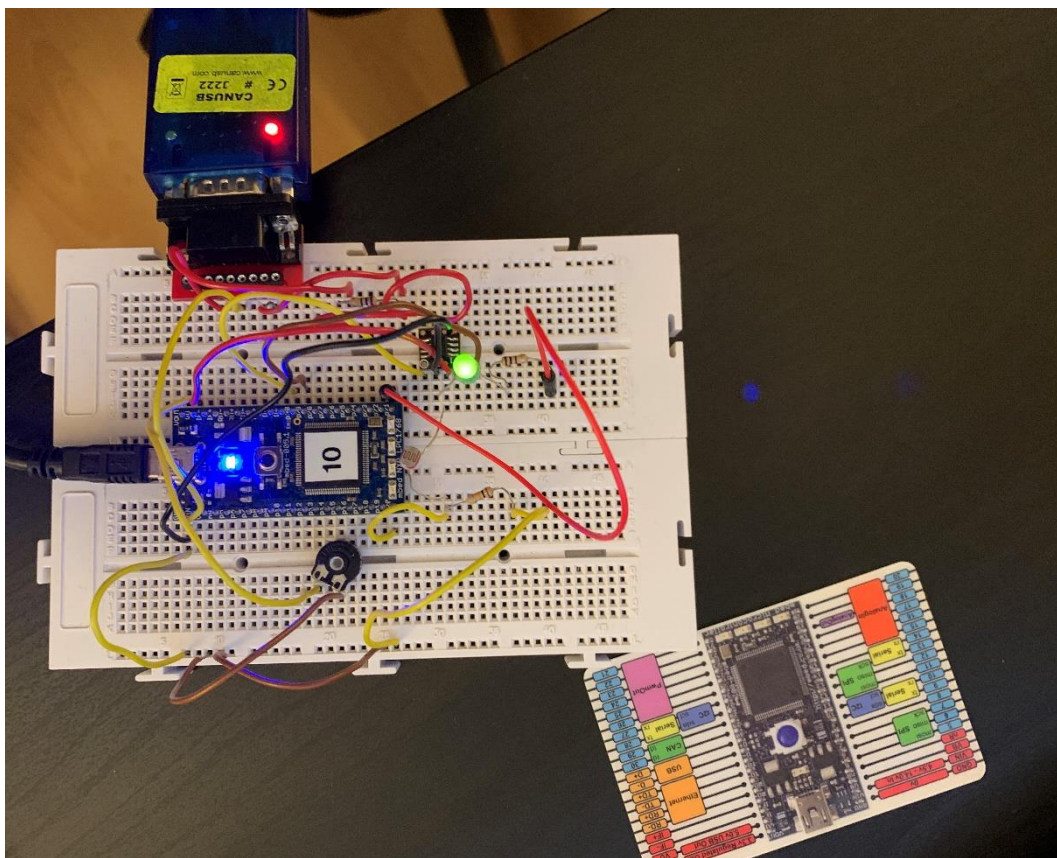


## Sistemes Encastats i Ubics

### Laboratori 4 – Comunicació amb Bus CAN



*Carlota Catot*

*Miguel Antúnez*

Comentarios:

## Parte CAN

Hemos tenido problemas con la comunicación desde la app de c# por lo que hemos tenido que hardcodear las constantes PID.

Código:

```
void read() {  
  
    CANMessage msg;  
    printf("read() - %d\n", can.read(msg));  
    printf("Message received: %d\n", msg.data[0]);  
  
    if(can.read(msg)){  
        Kp = msg.data[0];  
        Ki = msg.data[1];  
        Kd = msg.data[2];  
    }  
}
```

## Parte PID

Hemos utilizado una librería para la parte del controlador PID

<https://os.mbed.com/users/aberk/code/PID/docs/tip/classPID.html>

Para determinar las constantes hemos probado diferentes combinaciones ayudándonos con putty para imprimir por pantalla los valores de luminosidad y la salida del controlador PID nos hemos quedado con la siguiente combinación:

```
float Kp = 1.0;    // Proportional gain  
float Ki = 0.0;    // Integral gain  
float Kd = 0.0;    // Differential gain
```

El código completo de toda la práctica:

```
#include "mbed.h"  
#include "PID.h"  
  
DigitalOut myled(LED1);  
  
Ticker ticker;  
DigitalOut led1(LED1);  
DigitalOut led2(LED2);  
CAN can(p30, p29);  
Serial pc(USBTX, USBRX, 115200);  
  
PwmOut led(p21);  
AnalogIn potentiometer(p16);  
AnalogIn ldr(p20);
```

```

float Kp = 1.0;      // Proportional gain
float Ki = 0.0;      // Integral gain
float Kd = 0.0;      // Differential gain

//Kc, Ti, Td, interval
PID controller(Kp, Ki, Kd, 0.1);

void read() {

    CANMessage msg;
    printf("read() - %d\n", can.read(msg));
    printf("Message received: %d\n", msg.data[0]);

    if(can.read(msg)){
        Kp = msg.data[0];
        Ki = msg.data[1];
        Kd = msg.data[2];
    }
}

int main() {

    can.frequency(500000);
    ticker.attach(&read, 1);

    controller.setInputLimits(0.0, 3.3);
    controller.setOutputLimits(0.0, 1.0);
    controller.setBias(0.3);
    controller.setMode(1);
    controller.setSetPoint(1.7);

    while(1) {

        led.write(potentiometer.read());
        pc.printf("potentiometer: %.6f\n", potentiometer.read());
        pc.printf("brightness: %.6f\n", ldr.read());

        controller.setProcessValue(ldr.read());
        led.write(controller.compute());
        pc.printf("PID value %.6f\n", controller.compute());

        wait(0.1);

    }
}

```