Universidad del Valle de Guatemala
Facultad de Ingeniería
Electrónica Digital 2
Mini proyecto 2
Jonatan Chocón, 18420

MCU

```
función Read_esp32()
        Si UART_DATA_Ready es verdadero
                Mientras RCIF es falso realizar loop infinito
                Establecer PORTD a RCREG;
                __delay_ms(100);

función Write_esp32()
        Establecer TXREG a dato_a_enviar;
        Mientras TRMT es falso realizar loop infinito
        __delay_ms(100);

función Read_sensor()
        call I2C_Master_Start();
        call I2C_Master_Stop();

configuración setup()
        establecer TRISD a 0;
        establecer PORTD  a 0;
        establecer TRISC  a 0;
        establecer PORTC a 0;

principal main()
        call setup();
        call I2C_Master_Init()
        UART_Init();
        loop principal
                call Read_sensor();
                call Write_esp32();
                call Read_esp32();
```

UART_Library

```
función UART_Init()
        establecer SPBRG a 25;
        establecer BRGH a 1;
        establecer SPBRGH a 0;
        establecer SYNC a 0;
        esatablecer SPEN a 1;
        establecer TX9 a 0;
        establecer TRISC7 a 1;
        establecer TRISC6 a 0;
        establecer CREN a 1;
        establecer TXEN a 1;
```

```
función UART_DATA_Ready()
        retornar RCIF;
```

I2C_Library

```
función I2C_Master_Init()
        establecer SSPADD;
        establecer SSPSTAT a 0;
        establecer SSPCON  a 0b00101000;
        establecer TRISC3 a 1;
        establecer TRISC4 a 1;

función I2C_Master_Wait()
        Mientras (SSPSTAT & 0b00000100) || (SSPCON2 & 0b00011111)
                // loop infinito

función I2C_Master_Start()
        call I2C_Master_Wait();
        establecer SEN a 1;

función I2C_Master_Stop()
        call I2C_Master_Wait();
        establecer PEN a 1;
```

pubsup_adafruit

```
establecer Adafruit_Feed *P1 = io.feed(“P1”);
establecer Adafruit_Feed *P1 = io.feed(“P2”);
establecer Adafruit_Feed *SENSOR = io.feed(“SENSOR”);

principal configuracion_principal
        Serial.begin(115200);
        establecer parametros de Serial2;
        call io.connetc(); // conectar con io.adafruit.com
        asignar un onMessage a P1
        asignar un onMessage a P2
        asignar un onMessage a SENSOR

        Mientras io.status() sea menor que AIO_CONNECTED
                Imprimir io.StatusText();
                delay(500);

        P1->get();
        P2->get();
        SENSOR->get();
```

```
Loop principal
        Mientras Serial2.available sea verdadero
                Establecer Read_pic a Serial2.read();
                Write_pic = State_p1 + State_p2;
                Call Serial2.Write y enviar Write_pic;

        Si millis() es mayor que (lastUpdate + IO_LOOP_DELAY)
                Guardar Read_pic en el feed SENSOR;
                Establecer lastUpdate a millis();

función handleSensor()
        imprimir “received <-”;
        imprimir value del feed SENSOR;

función handleP1()
        Si data->isTrue()
                Establecer state_p1 a 1;
        Si no
                Establecer state_p1 a 0;

función handleP2()
        Si data->isTrue()
                Establecer state_p2 a 2;
        Si no
                Establecer state_p2 a 0;
```