

Ultrasonic LED Bar representation (Sprint 4)

MINF UDL 20-21

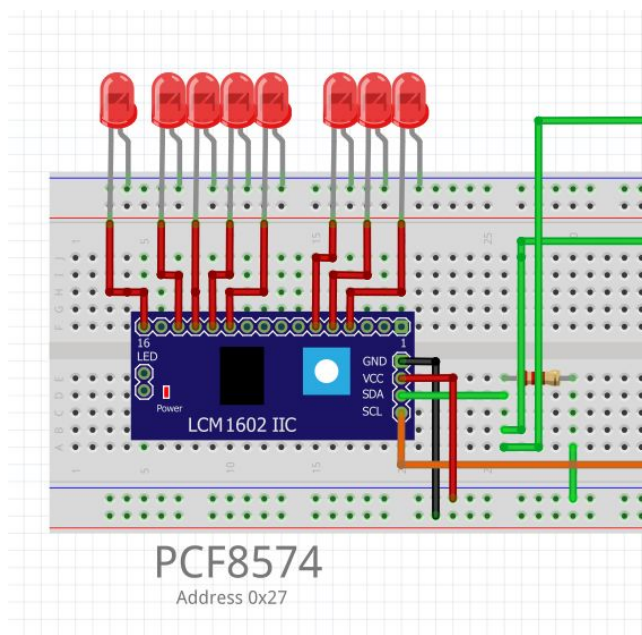
Ubiquitous and embedded systems

Team 1

1. Objective

- a. The objective is to work with the PCF8574 to turn on LEDs on a LED Bar to represent the level of the container, it receives the distance from the sensor and makes the calculation to display the correct amount of LED's turned on.

2. Wiring



In this case the PCF is connected to the Raspberry via the SDA and SCL lines (with a transistor at pullup of these lines).

3. Code (ChibiOS)

- a. Considerations: In order to the Raspberry to command the PCF, we must follow these steps:

- i. Initialize the PCF

- 1. We send a 0x00 to the PCF to activate.

```
i2cled_init(pcf_address, 0x00);
```

- ii. Send the information to the PCF

- 1. Now we have to send to the PCF 2 things: One is its own address + 0 (to indicate it's a write operation) + the data in which will tell which pins will be lighted up.

- a. The address we are using

```
#define pcf_address 0x27 //device  
address  
#define pcf_address_write 0x4E // pcf  
address + 0 bit write
```

- b. The data we are sending, its 8 bits

```
0b11110110
```

- c. Send both information in a array of size 2

```
request[0] = pcf_address_write;  
request[1] = data;  
  
msg_t status = i2cMasterTransmit(  
    &I2C0, device_address, request, 2,  
    NULL, 0);
```

- b. Full code:

```
static WORKING_AREA(waThread_PCF, 128);  
static msg_t Thread_PCF(void *p)  
{  
    (void)p;  
    chRegSetThreadName("PCF");  
    chThdSleepMilliseconds(2000);  
    int distanceHandler = 0;  
  
    while (TRUE)  
    {  
        chBSemWait(&smph);
```

```

    palSetPad(ONBOARD_LED_PORT, ONBOARD_LED_PAD);

    distanceHandler = handleDistance(distance);
    pinOut = (uint8_t)handleMeasure(distanceHandler);

    //i2cMasterTransmit(&I2C0, pcf_address_write, (uint8_t)pinOut,
sizeof(pinOut), NULL, 0);
    i2cled_write(pcf_address, pcf_address_write, pinOut);

    //lcdPrintf(92, 17, "%u", pinOut);

    palClearPad(ONBOARD_LED_PORT, ONBOARD_LED_PAD);
    aux_counter += 1;
    aux_pcf += 1;
    //temperature += 3;
    //humidity += 5;

    if (aux_pcf == 8)
        aux_pcf = 0;
    if (aux_counter % 15 == 0)
    {
        temperature = 6;
        humidity = 7;
    }
    chThdSleepMilliseconds(2000);
    chBSemSignal(&smph);
}

return 0;
}
static msg_t i2cled_write(uint8_t device_address,
                          uint8_t register_address,
                          uint8_t data)
{
    uint8_t request[2];
    request[0] = register_address;
    request[1] = data;

    msg_t status = i2cMasterTransmit(
        &I2C0, device_address, request, 2,
        NULL, 0);
    chThdSleepMilliseconds(50);

    if (status != RDY_OK)
        //chprintf((BaseSequentialStream *)&SD1, "Error while writing to
i2cled: %d\r\n", status);

```

```
    return status;
}

static void i2cled_init(uint8_t device_address, uint8_t dirmask)
{
    msg_t status = i2cled_write(device_address,
                                pcf_address_write, // direction register.
                                dirmask);

    chThdSleepMilliseconds(50);

    if (status != RDY_OK)
        //chprintf((BaseSequentialStream *)&SD1, "Error while setting direction
        mask: %d\r\n", status);
}
```

4. Final result

Reference video: <https://www.youtube.com/watch?v=WJ3wlqHZit0>

