

Interfaçage des télémètres

Pour mesurer la distance entre le chariot et la carotte sur Z, ou pour mesurer la position par rapport à la butée sur l'axe X, on utilise des télémètres laser à temps de vol (TOF = *time of flight*) de la famille VL53L (0 ou 1) X, connectés en I2C.

Initialement, nous imaginions qu'un simple hub suffirait : les breakout board des TOF étant un VL53L0X et l'autre VL53L1X de deux fabricants différents, on espérait deux adresses différentes. Las : ils sont tous les deux à #0x029 (bah oui, la puce reste un ST Micro de la même ligne, [explication sur le forum de ST Micro \[archive\]](#)). La procédure permettant de forcer un changement d'adresse logiciellement en passant par un signal supplémentaire XSHUT semble lourde : souder à nouveau, câbler à nouveau, refaire le boîtier du M5 pour rendre accessible deux GPIO...

Il semble plus facile de remplacer le hub par un "multiplexeur", par exemple à base de TCA9548A, de chez Seeedstudio qu'on a en stock. Problème : la [documentation Seeedstudio](#) et la bibliothèque sont relativement mal foutues (même les commentaires du début de code sont pleins de coquilles et illisibles). La [documentation de l'équivalent chez Adafruit](#) est bien meilleure (comme d'habitude ! ☹️). Mais est-elle transposable à 100% ?

Avec le code proposé par Adafruit on obtient le programme suivant (qui fonctionne) pour lire les valeurs des 2 capteurs :

```
#include <Wire.h>
#include "M5Unified.h"
#include <VL53L0X_mod.h>          // Library for VL53L0X sensor
#include <SparkFun_VL53L1X.h>    // Library for VL53L1X sensor (adjust if needed)

// I2C address for the TCA9548A multiplexer
#define TCAADDR 0x70

// Provided function to select a channel on the multiplexer
void tcselect(uint8_t i) {
  if (i > 7) return;
  Wire.beginTransmission(TCAADDR);
  Wire.write(1 << i);
  Wire.endTransmission();
}
```

```

// Create sensor objects:
// - sensorVL0X will be on multiplexer channel 0
// - sensorVL1X will be on multiplexer channel 1
VL53L0X_mod sensorVL0X;
SFEVL53L1X sensorVL1X;

void setup() {
  // Initialize M5Stack and I2C
  M5.begin();
  Wire.begin();
  Serial.begin(115200);
  delay(1000); // Allow some time for sensor power-up

  // ----- Initialize VL53L0X on multiplexer channel 0 -----
  tcselect(0);
  if (!sensorVL0X.init()) {
    Serial.println("Failed to initialize VL53L0X on channel 0");
    while (1);
  }
  sensorVL0X.setTimeout(500);
  sensorVL0X.startContinuous();
  Serial.println("VL53L0X initialized on channel 0");

  // ----- Initialize VL53L1X on multiplexer channel 1 -----
  tcselect(1);
  if (sensorVL1X.begin() != 0) { // 'begin()' for VL53L1X initialization
    Serial.println("Failed to initialize VL53L1X on channel 1");
    while (1);
  }
  sensorVL1X.startRanging(); // Start continuous ranging (or single-shot if preferred)
  Serial.println("VL53L1X initialized on channel 1");
}

void loop() {
  uint16_t distanceVL0X, distanceVL1X;

  // ----- Read from VL53L0X on channel 0 -----
  tcselect(0);
  distanceVL0X = sensorVL0X.readRangeContinuousMillimeters();

```

```
// ----- Read from VL53L1X on channel 1 -----  
tcselect(1);  
distanceVL1X = sensorVL1X.getDistance();  
  
// ----- Output the results -----  
Serial.print("VL53L0X (Ch 0): ");  
Serial.print(distanceVL0X);  
Serial.print(" mm \t | \tVL53L1X (Ch 1): ");  
Serial.print(distanceVL1X);  
Serial.println(" mm");  
  
delay(100);  
}
```

Revision #4

Created 14 February 2025 07:05:17 by Christian Simon

Updated 15 February 2025 10:37:10 by Stephane Muller