

HOS04 : INSTALLATION MODULE ASANGLE (NODE-RED)

1. Objectif.....	1
2. Prérequis.....	2
3. Compétences.....	2
4. Activités.....	2
4.1 Installation matériel.....	2
4.2 Installation logicielle.....	3
5. Mesure et validation.....	3
6. Complément : étude du schéma détaillé.....	4
6.1 Étude du module PODOLU LSM303D.....	4
6.2 Étude du M5StickCPlus.....	4
7. Ressources.....	5
7.1 Schéma module ASangle.....	5
7.2 PCB module ASangle.....	5
7.3 Programmes.....	6
8. Correction.....	11

1. Objectif

On souhaite connaître l'inclinaison et l'azimut du panneau solaire afin d'optimiser son rendement.

Les mesures seront envoyées vers le logiciel rapidSCADA. (voir le TP AS40 rapidSCADA).

2. Prérequis

Utilisation de l'IDE Arduino avec le M5StickCplus (cf https://docs.m5stack.com/en/quick_start/m5stickc_plus/arduino)

Lire le TP AS40 rapidSCADA.

3. Compétences

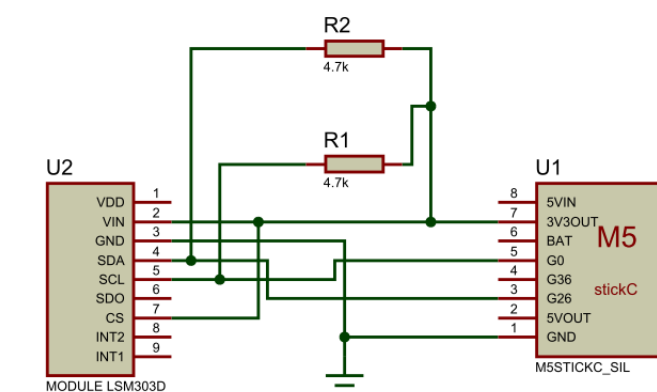
4. Activités

4.1 Installation matériel

Le module est fournit en kit :

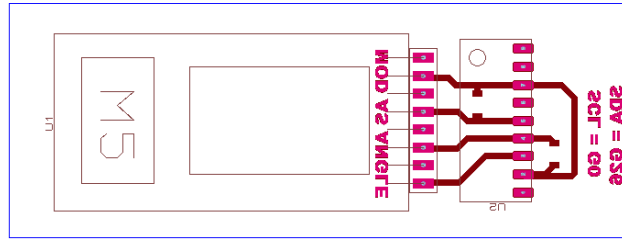
la boîte contient : 1 x M5StickCplus vierge, 1 x LSM303D, 1 x PCB noté 'mod ASangle'

Le schéma et le PCB sont donnés dans les ressources.



GPIO36 JAMAIS une sortie

FILE NAME:	M5stickC_ASangle.pdsprj	DATE:	06/07/2023
DESIGN TITLE:	M5stickC_ASangle.pdsprj	PAGE:	01
PATH:	E:\Data\enCours\sts1sne\cartes\carte 2023 ASangle\	TIME:	09:44:28
BY:	@AUTHOR	REV:	@REV



4.1.1 Assemblage des éléments

L'alimentation se fait par la batterie intégrée dans le M5StickCPlus. Elle est rechargée par le port USB-C qui permet aussi la programmation.

- Dessiner le connecteur du M5 soyez attentif à l'alimentation.
- Dessiner le module LSM303D, soyez attentif à l'alimentation.
- Assembler correctement les éléments en étant vigilant aux broches d'alimentation.

4.2 Installation logicielle

Le M5 est fourni vierge.

Pour le programmer avec l'IDE , vous suivrez la procédure fournie par le fabricant (voir Prérequis)

4.2.1 Récupérer le programme : ASangle.ino

4.2.2 Ouvrir le programme avec l'IDE Arduino (v1.8)

4.2.3 Comprendre le programmer

4.2.4 Modifier le programme afin de l'adapter à votre configuration

SSID, IP, brochage

5. Mesure et validation

Les données sont envoyées par le programme vers le serveur SCADA que vous avez configuré dans le programme.

Afin de visualiser les échanges de données, on utilise le logiciel Wireshark.

5.1.1 Envoyer les données du module ASangle vers un PC ToutEnUn Wifi contenant Wireshark et le serveur Node-red. (voir TP HOS03)

5.1.2 Capturer les échanges

5.1.3 Quel protocole est utilisé pour envoyer les données, entre le module ASangle et le serveur Node-red?

5.1.4 Isoler la trame et retrouver les mesures effectuées.

5.1.5 Avec node-red, identifier la trame de demande du protocole (vous pouvez vous aider de la doc. Technique du protocole.)

5.1.6 Comparer avec les échanges capturés.

5.1.7 Faire valider par l'enseignant.

6. Complément : étude du schéma détaillé

6.1 Étude du module PODOLU LSM303D

A partir de la documentation technique du module .

6.1.1 Étudier le schéma en repérant les différentes fonctions.

Alimenter, adapter, capter .

6.2 Étude du M5StickCPlus

Étudier le schéma présent dans la documentation technique.

6.2.1 Quelle alimentation est utilisée ?

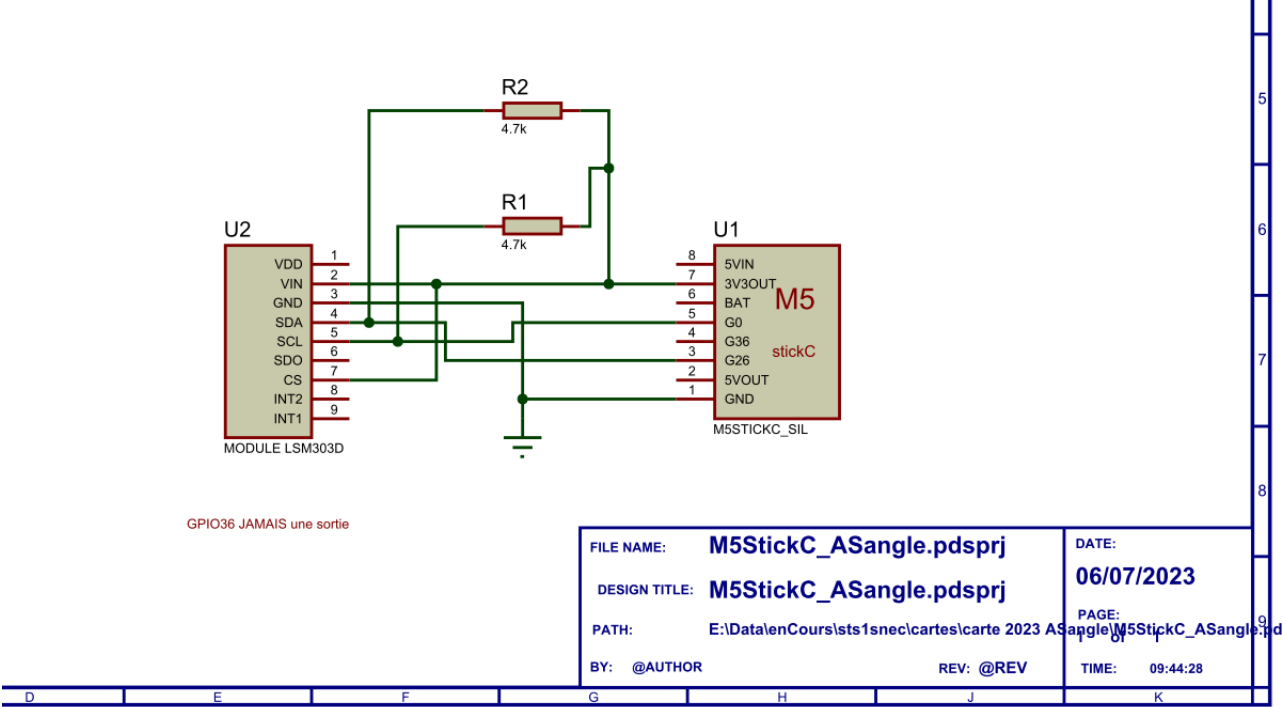
6.2.2 Quels capteurs sont présents dans le M5 ?

6.2.3 Quels sont les actionneurs présents dans le M5 ?

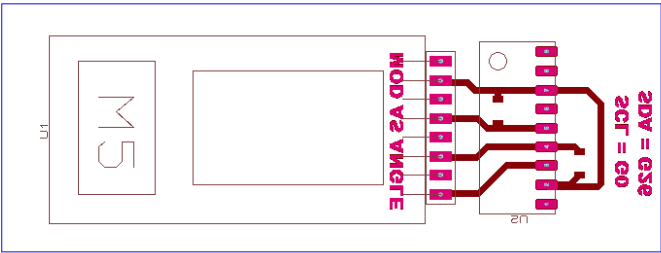
6.2.4 Quelles sont les caractéristiques du LCD ?

7. Ressources

7.1 Schéma module ASangle



7.2 PCB module ASangle



7.3 Programmes

7.3.1 M5_ASangle.ino

```
/*
 * AS angle
 * Module à placer sur le panneau solaire
 * Le module mesure
 * 1.l'inclinaison sur l'axe Y (à vérifier)
 * 2.l'orientation de 0 à 360°
 * Les mesures sont envoyés vers rapidSCADA en modbus
 * Un page web embarqué permet de voir les mesures en direct //option
 * Un affichage LCD affiche : les mesures et l'adresse IP.
 * La configuration se fera par reprogrammation ou/et par la page web
 */

#include <M5StickCPlus.h>
#include <Wire.h>
#include <LSM303.h>
#include <WiFi.h>
#include <ModbusIP_ESP8266.h>

LSM303 compass;

//modbus
const int Azimut_HREG = 200;
const int Elevation_HREG=201;

//ModbusIP object
ModbusIP mb;
IPAddress staticIP(192, 168, 0, 60);
IPAddress subnet(255, 255, 255, 0);
IPAddress gateway(192, 168, 0, 254);

long ts;
```

```
/*  
*/  
*/  
  
void setup() {  
  Serial.begin(115200);  
  
  //WIFI  
  
  if (WiFi.config(staticIP, gateway, subnet) == false) {  
    Serial.println("Configuration failed.");  
  }  
  
  WiFi.begin("votre SSID", "pass du SSID");  
  while (WiFi.status() != WL_CONNECTED) {  
    delay(500);  
    Serial.print(".");  
  }  
  
  Serial.println("");  
  Serial.println("WiFi connected");  
  Serial.println("IP address: ");  
  Serial.println(WiFi.localIP());  
  
  //finWIFI  
  
  // initialize the M5StickC object  
  M5.begin();  
  M5.Lcd.setRotation(3);  
  // text print  
  M5.Lcd.fillScreen(BLACK);  
  M5.Lcd.setCursor(0, 10);  
  M5.Lcd.setTextColor(WHITE);  
  M5.Lcd.setTextSize(2);  
  M5.Lcd.printf("ASangle");  
  //
```

HOS04 : installation module ASangle (node-red)

```
//Modbus

    mb.server();    //Start Modbus IP
    // Add SENSOR_IREG register - Use addHreg()
    mb.addHreg(Azimut_HREG);
    mb.addHreg(Elevation_HREG);

//fin Modbus

//compass
    Wire.setPins(26,0);//attribut les broche I2C (bool setPins(int sdaPin, int
sclPin);)
    Wire.begin();//changer les port I2C.
    compass.init();
    compass.enableDefault();

    /*
    Calibration values; the default values of +/-32767 for each axis
    lead to an assumed magnetometer bias of 0. Use the Calibrate example
    program to determine appropriate values for your particular unit.
    */
    compass.m_min = (LSM303::vector<int16_t>){-32767, -32767, -32767};
    compass.m_max = (LSM303::vector<int16_t>){+32767, +32767, +32767};
//fin compass
    ts = millis();
} //fin setup

/*****/
/*****/
void loop() {

int valAzimut, valElevation;

    //compass
    compass.read();

    /*
    When given no arguments, the heading() function returns the angular
```

difference in the horizontal plane between a default vector and north, in degrees.

The default vector is chosen by the library to point along the surface of the PCB, in the direction of the top of the text on the silkscreen. This is the +X axis on the Pololu LSM303D carrier and the -Y axis on the Pololu LSM303DLHC, LSM303DLM, and LSM303DLH carriers.

To use a different vector as a reference, use the version of heading() that takes a vector argument; for example, use

```
compass.heading((LSM303::vector<int>){0, 0, 1});
```

to use the +Z axis as a reference.

```
*/  
float heading = compass.heading();  
int angleX = (int)compass.a.x;  
int angleY = (int)compass.a.y;  
int angleZ = (int)compass.a.z;  
  
int16_t inclination = atan2(angleY, sqrt(angleX * angleX + angleZ * angleZ)) *  
180.0 /PI;  
valAzimut = round(heading) ;  
valElevation = inclination;  
Serial.println(heading);  
  
//affichage vers LCD  
IPAddress ipAdr;  
ipAdr = WiFi.localIP();  
M5.Lcd.setCursor(120,20); M5.Lcd.setTextSize(1); M5.Lcd.setTextColor(BLUE);  
M5.Lcd.printf("IP : %u.%u.%u.%u\n\r",ipAdr[0],ipAdr[1],ipAdr[2],ipAdr[3]);  
  
M5.Lcd.setCursor(10, 40);  
M5.Lcd.setTextColor(RED);  
M5.Lcd.setTextSize(3);  
M5.Lcd.printf("Bous:%6d",(int)heading);  
M5.Lcd.setCursor(10, 80);
```

HOS04 : installation module ASangle (node-red)

```
M5.Lcd.setTextColor(RED);
M5.Lcd.setTextSize(3);
M5.Lcd.printf("Angle:%6d",(int)inclination);
delay(1000);

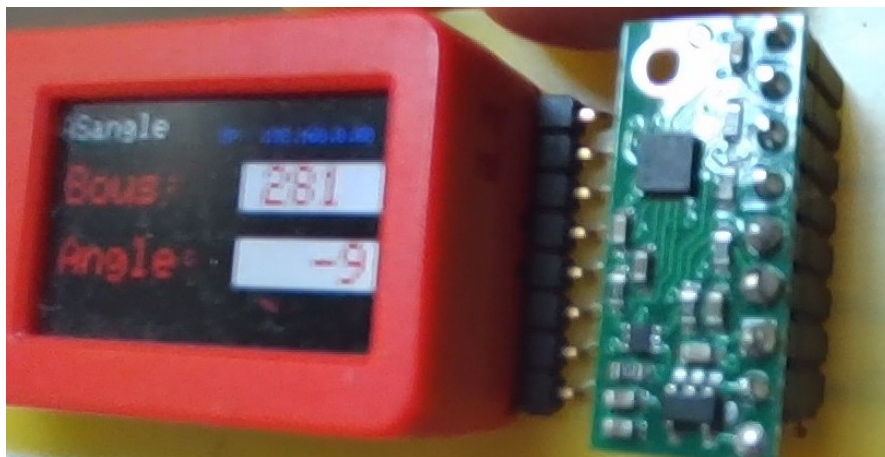
    M5.Lcd.fillRect(140,39,90,25,WHITE);//M5.Lcd.fillRect(int16_t x, int16_t y,
int16_t w, int16_t h, uint16_t color);
    M5.Lcd.fillRect(140,79,90,25,WHITE);//M5.Lcd.fillRect(int16_t x, int16_t y,
int16_t w, int16_t h, uint16_t color);

//modbus
    //ModBus Call once inside loop() - all magic here
    mb.task();
    //Read each two seconds
    if (millis() > ts + 2000) {
        ts = millis();
        mb.Hreg(Azimut_HREG,valAzimut);
        mb.Hreg(Elevation_HREG,valElevation);

        Serial.println("envoi modbus");
    }
    delay(10);
//fin modbus

}
```

8. Correction



HOS04 : installation module ASangle (node-red)

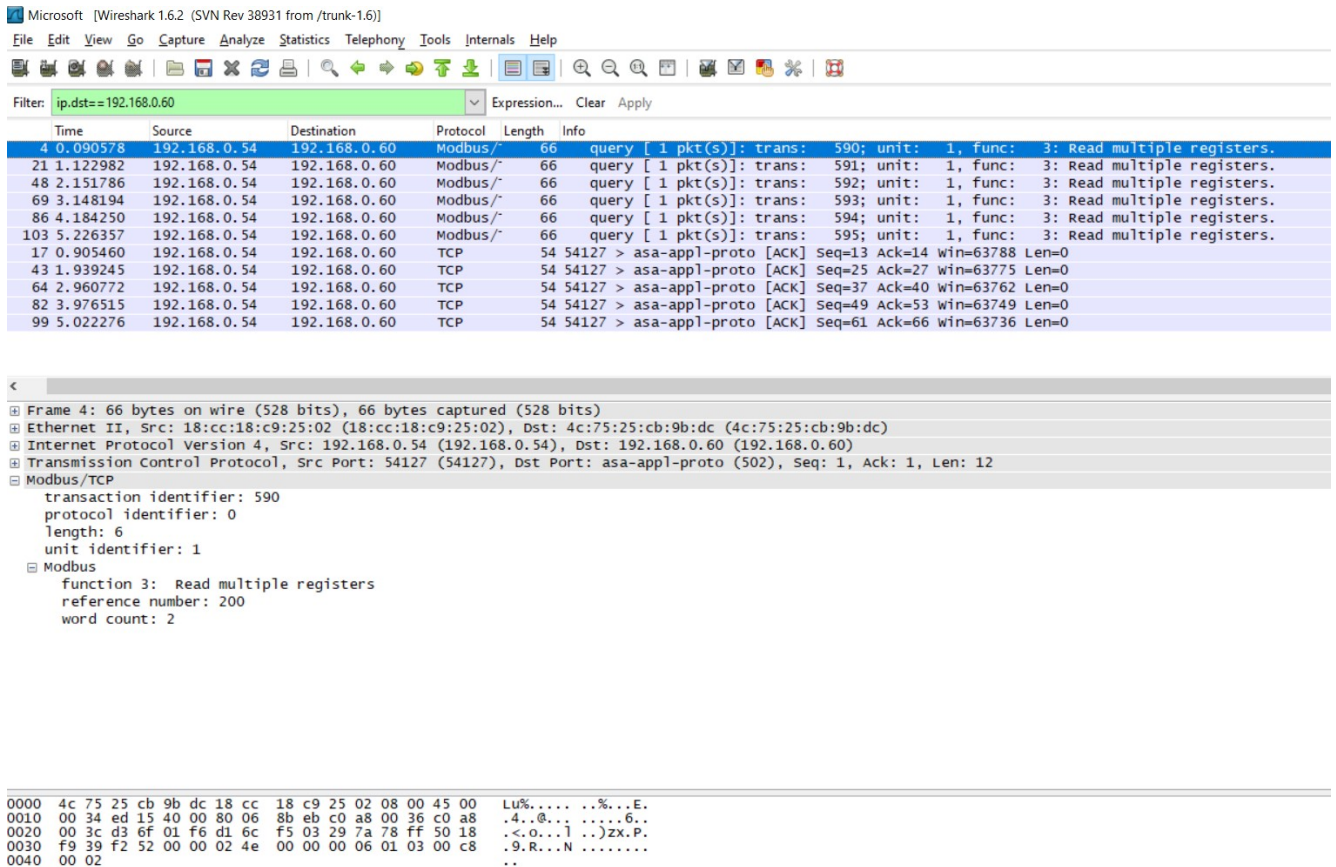


Figure 1: Wireshark : requête vers module ASangle à partir de rapidSCADA

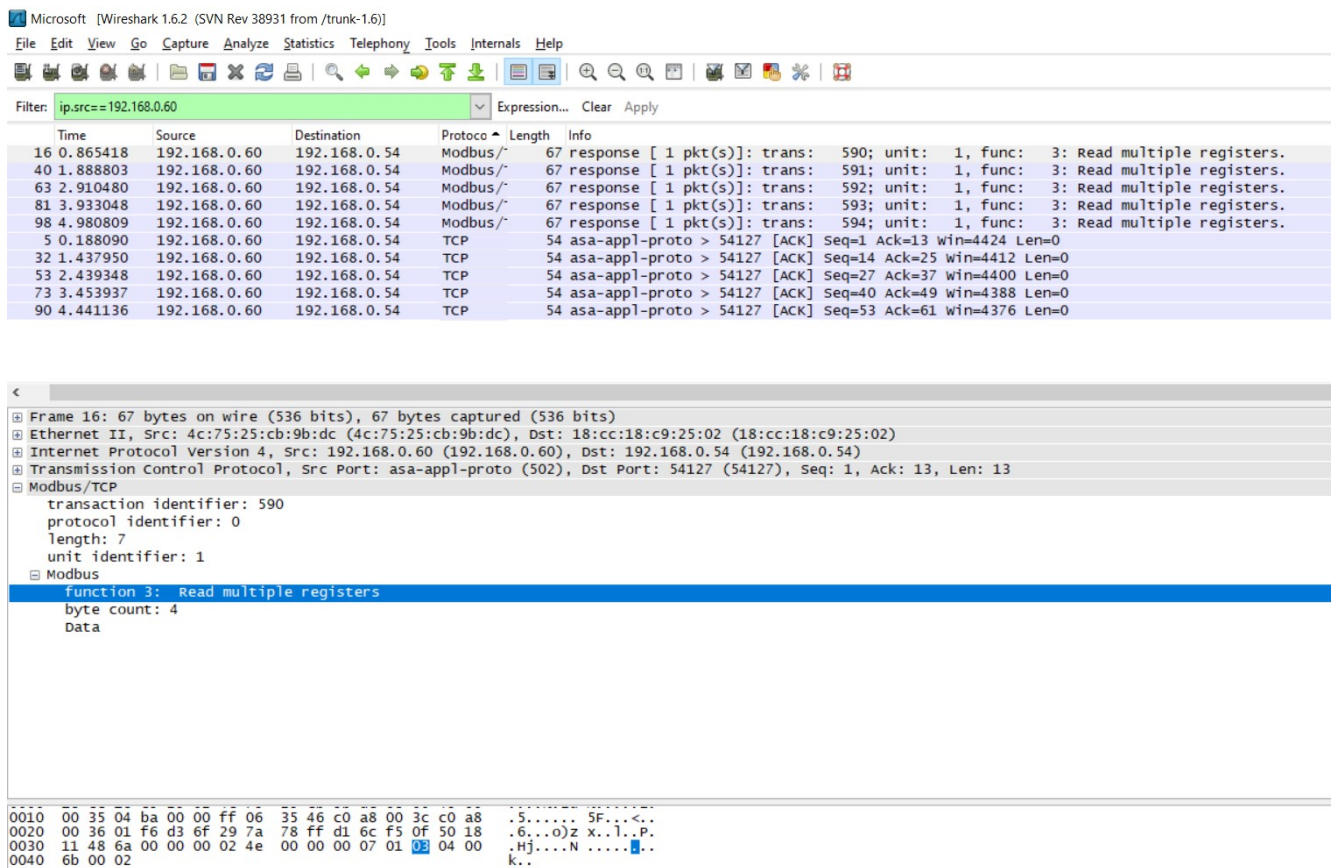


Figure 2: Wireshark : réponse du module ASangle vers rapidSCADA

On a en vrai (voir photo) : boussole = 107 = 0x006b et inclinaison = 2 = 0x0002.

On retrouve bien nos données.

On voit dans la requête l'adresse modbus : 200 = 0x00C8.