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. Correction1	1

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 <u>https://docs.m5stack.com/en/quick_start/m5stickc_plus/arduino</u>) Lire le TP AS40 rapidSCADA.

3. Compétences

4. Activités

4.1 Installation matériel

Le module est fournit en kit :

la boite contient : 1 x M5StickCplus vierge, 1 x LSM303D, 1 x PCB noté 'mod ASangle' Le schéma et le PCB sont donnés dans les ressources.





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 fournit 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	
*/	
<pre>#include <m5stickcplus.h></m5stickcplus.h></pre>	
<pre>#include <wire.h></wire.h></pre>	
<pre>#include <lsm303.h></lsm303.h></pre>	
<pre>#include <wifi.h></wifi.h></pre>	
<pre>#include <modbusip_esp8266.h></modbusip_esp8266.h></pre>	
LSM303 compass;	
//modbus	
const int Azimut_HREG = 200;	
const int Elevation_HREG=201;	
//ModbusiP object	
ModbusIP mb;	
IPAdaress staticIP(192, 168, 0, 60);	
IPAddress subnet(255, 255, 255, 0);	
IPAddress gateway(192, 168, 0, 254);	
long to:	

```
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
```

HOS04 : installation module ASangle (node-red)

```
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)

Microsoft [Wireshark 1.6.2 (SVN Rev 38931 from /trunk-1.6)]

<u>File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help</u>

		0	0		F		×	2			4			T	₽			10	Q	1	**		Y	.	×	
--	--	---	---	--	---	--	---	---	--	--	---	--	--	---	---	--	--	----	---	---	----	--	---	----------	---	--

Filter:	ip.dst==192.16	8.0.60		 Expression 	ion	Clear Ap	ply												
	Time	Source	Destination	Protocol Lengt	th In	fo													
4	0.090578	192.168.0.54	192.168.0.60	Modbus/-	66	query	[1	<pre>pkt(s)]:</pre>	trans:	590;	unit:	1,	func:	3:	Read	multi	ple r	egisters	
21	1.122982	192.168.0.54	192.168.0.60	Modbus/-	66	query	[1	<pre>pkt(s)]:</pre>	trans:	591;	unit:	1,	func:	3:	Read	multi	ple r	egisters	
48	2.151786	192.168.0.54	192.168.0.60	Modbus/*	66	query	[1	pkt(s)]:	trans:	592;	unit:	1,	func:	3:	Read	multi	ple r	egisters	
69	3.148194	192.168.0.54	192.168.0.60	Modbus/*	66	query	[1	<pre>pkt(s)]:</pre>	trans:	593;	unit:	1,	func:	3:	Read	multi	ple r	egisters	
86	4.184250	192.168.0.54	192.168.0.60	Modbus/-	66	query	[1	pkt(s)]:	trans:	594;	unit:	1,	func:	3:	Read	multi	ple r	egisters	
103	5.226357	192.168.0.54	192.168.0.60	Modbus/*	66	query	[1	pkt(s)]:	trans:	595;	unit:	1,	func:	3:	Read	multi	ple r	egisters	
17	0.905460	192.168.0.54	192.168.0.60	TCP	54 5	4127 >	asa-a	app1-prote	D [ACK]	Seq=13	Ack=14	Win=	63788	Len=0	1				
43	1.939245	192.168.0.54	192.168.0.60	TCP	54 5	4127 >	asa-a	app1-prote	D [ACK]	Seq=25	Ack=27	Win=	63775	Len=0	1				
64	2.960772	192.168.0.54	192.168.0.60	TCP	54 5	4127 >	asa-a	app1-prote	D [ACK]	Seq=37	Ack=40	Win=	63762	Len=0	l.				
82	3.976515	192.168.0.54	192.168.0.60	TCP	54 5	4127 >	asa-a	app1-prote	D [ACK]	Seq=49	Ack=53	Win=	63749	Len=0	1				
99	5.022276	192.168.0.54	192.168.0.60	TCP	54 5	4127 >	asa-a	appl-prot	D FACK]	Seq=61	Ack=66	Win=	63736	Len=0	1				

<

- Frame 4: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) Ethernet II, Src: 18:cc:18:c9:25:02 (18:cc:18:c9:25:02), Dst: 4c:75:25:cb:9b:dc (4c:75:25:cb:9b:dc) Internet Protocol Version 4, Src: 192.168.0.54 (192.168.0.54), Dst: 192.168.0.60 (192.168.0.60) Transmission Control Protocol, Src Port: 54127 (54127), Dst Port: asa-appl-proto (502), Seq: 1, Ack: 1, Len: 12 Hodbuc Crop.

1	Modbus/TCP	
	transaction identifier: 590	
	protocol identifier: 0	
	length: 6	
	undt delentifiers 1	

unit identifier: 1 ⊡ Modbus

odbus function 3: Read multiple registers reference number: 200 word count: 2

0000	4c 7	5 25	cb	9b	dc	18	CC	18	c 9	25	02	08	00	45	00	Lu%%E.
0010	00 3	4 ed	15	40	00	80	06	8b	eb	c 0	a8	00	36	c0	a8	.4@6
0020	00 3	c d3	6f	01	f6	d1	6c	f5	03	29	7a	78	ff	50	18	.<.o])zx.P.
0030	f9 3	9 f2	52	00	00	02	4e	00	00	00	06	01	03	00	c8	.9.RN
0040	00 0	2														

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

Microsoft	Microsoft [Wireshark 1.6.2 (SVN Rev 38931 from /trunk-1.6)]															
<u>File Edit</u>	View <u>G</u> o (Capture Analyze S	tatistics Telephony <u>T</u> o	ols Internals H	elp											
		🖻 🖬 🗙 😂 🌡	∃ Q, ⇔ ⇔ ⇔	₮ 🕹 🔳 🛛	. ⊕, Q	0, 🖭 🕯	ă 🗹 📒	8 %	12							
Filter: ip.src	==192.168.0.6	.60		✓ Expressi	on Clear	Apply										
Time	So	ource	Destination	Protoco - Lengt	h Info											
16 0.86	5418 1	92.168.0.60	192.168.0.54	Modbus/	67 respons	e [1 pkt	(s)]: t	rans:	590;	unit:	1, func	: 3:	Read n	nultiple	registers	i.
40 1.88	88803 1	92.168.0.60	192.168.0.54	Modbus/-	67 respons	e [1 pkt	(s)]: t	rans:	591;	unit:	1, func	: 3:	Read n	nultiple	registers	
63 2.91	.0480 1	92.168.0.60	192.168.0.54	Modbus/-	67 respons	e [1 pkt	(s)]: t	rans:	592;	unit:	1, func	: 3:	Read n	nultiple	registers	i.
81 3.93	3048 1	92.168.0.60	192.168.0.54	Modbus/*	67 respons	e [1 pkt	(s)]: t	rans:	593;	unit:	1, func	: 3:	Read n	nultiple	registers	i.
98 4.98	80809 1	92.168.0.60	192.168.0.54	Modbus/-	67 respons	e [1 pkt	(s)]: t	rans:	594;	unit:	1, func	: 3:	Read n	nultiple	registers	s.
5 0.18	88090 1	92.168.0.60	192.168.0.54	TCP	54 asa-app	1-proto >	54127	[ACK]	Seq=1 A	ck=13 W	/in=4424	Len=0				
32 1.43	7950 1	92.168.0.60	192.168.0.54	TCP	54 asa-app	1-proto >	54127	[ACK]	Seq=14	Ack=25	win=4412	Len=0				
53 2.43	9348 1	92.168.0.60	192.168.0.54	TCP	54 asa-app	1-proto >	54127	[ACK]	Seq=27	Ack=37	win=4400	Len=0				
73 3.45	3937 1	92.168.0.60	192.168.0.54	TCP	54 asa-app	1-proto >	54127	[ACK]	Seq=40	Ack=49	win=4388	Len=0				
90 4.44	1136 1	92.168.0.60	192.168.0.54	TCP	54 asa-app	1-proto >	54127	[ACK]	Seq=53	Ack=61	win=4376	Len=0				

<	
🗄 Frame 16: 67 bytes on wire (536 bits), 67 bytes captured (536 bits)	
B Ethernet II, Src: 4c:75:25:cb:9b:dc (4c:75:25:cb:9b:dc), Dst: 18:cc:18:c9:25:02 (18:cc:18:c9:25:02)	
🟽 Internet Protocol Version 4, Src: 192.168.0.60 (192.168.0.60), Dst: 192.168.0.54 (192.168.0.54)	
🗄 Transmission Control Protocol, Src Port: asa-appl-proto (502), Dst Port: 54127 (54127), Seq: 1, Ack: 13, Len: 13	
Modbus/TCP	
transaction identifier: 590	
protocol identifier: 0	
length: 7	
unit identifier: 1	
Modbus	
function 3: Read multiple registers	
byte count: 4	
Data	
0010 00 35 04 ba 00 00 ff 06 35 46 c0 a8 00 3c c0 a8 5 5 5 5 c	
0020 00 36 01 f6 d3 6f 29 7a 78 ff d1 6c f5 0f 50 18 .6)z x1P.	
0030 11 48 6a 00 00 00 02 4e 00 00 00 7 01 03 04 00 .HjN	
0040 6b 00 02 k	

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.