COOLMAY HMI MT6070HA / ARDUINO CLONE / ARDUINO UNO and MODBUS RTU

0-Connecting the devices:

You need:

- -a COOLMAY MT6070H HMI: the master -2xMAX485 shields
- -an arduino uno board: the slave 3
- -an arduino clone board: the slave 1
- -an USBasp cable and 2xUSB cables





Pin No	Signal	Description			
Definition	of COM1 F	RS232 port			
2	RXD	Receive			
3	TXD	Send			
5	GND	Ground			
Definition of	Definition of COM1/COM2 RS485 port				
1	1 A				
6	B 4				







PART 1: HMI the MASTER, Clone the SLAVE1, UNO the SLAVE3 1-Arduino and softwares needed:

1-1- Install Arduino 1.8.13

<u>1-2- Go to https://github.com/Formator/SimpleModbus</u> for the modbus rtu lib downloading:

Formator / SimpleModbus					
<> Code (!) Issues (!) Pull requests	Actions III Projects	🕽 Security 🗠 Insights			
양 master 👻 양 1 branch 📀 0 tags		Go to file ↓ Code ▼			
Formator Update README.md		741d443 on 21 Jun 2016 🕚 5 commits			
SimpleModbusMaster	Renamed folders	5 years ago			
SimpleModbusMaster_DUE	Renamed folders	5 years ago			
SimpleModbusSlave	Softwere Serial	5 years ago			
SimpleModbusSlaveSoftwareSerial	Softwere Serial	5 years ago			
SimpleModbusSlave_DUE	Renamed folders	5 years ago			
	Initial commit	5 years ago			
🗅 README.md	Update README.md	5 years ago			
SimpleModbusMasterManual.pdf	Renamed folders	5 years ago			

README.md

Copy the libraries here:

- OneDrive
 Ce PC
 Bureau
 Occuments
 Arduino
 hardware
 - 🗸 📙 libraries
 - 🔉 📜 SimpleModbusMaster
 - > 📙 SimpleModbusMaster_DUE
 - > 🔋 SimpleModbusSlave
 - >] SimpleModbusSlave_DUE
 - >] SimpleModbusSlaveSoftwareSerial
 - > 📙 Schneider Electric
- > 📰 Images
- > 🁌 Musique

<u>1-3- Go to https://github.com/MCUdude</u> and download the 5 packs of boards: you need here the Minicore because I use an arduino clone board made of an Atmega328P

	Coverview Repositories 27 III Projects 🔗 Page	ckages
	Pinned	
	MightyCore Arduino hardware package for ATmega1284, ATmega644, ATmega324,	MegaCore Arduino hardware package for ATmega64, ATmega128, ATmega640,
	AImega324PB, AImega164, AImega32, AImega16 and AImega8535	Almega1280, Almega1281, Almega2560, Almega2561, Al90CAN32, Al90CAN64 and Al90CAN128
	● C++ ☆ 391 양 131	● C++ ☆ 191 ¥ 77
	🛱 MiniCore	📮 MicroCore
Hans MCUdude	Arduino hardware package for ATmega8, ATmega48, ATmega88, ATmega168, ATmega328 and ATmega328PB	An optimized Arduino hardware package for ATtiny13
Does hardware engineer stuff and	● C++ ☆ 501 😵 125	● Shell 🛱 324 😵 54
you find here is hobby stuff!	📮 MajorCore	📮 MegaCoreX
Follow	An Arduino hardware package for ATmega8515 and ATmega162	An Arduino hardware package for ATmega4809, ATmega4808, ATmega3209, ATmega3208, ATmega1609, ATmega1608, ATmega809 and ATmega808
유ኣ 351 followers · 6 following · ☆ 85	● C++ ☆ 25 😵 8	● C++ ☆ 75 왕 19

🖽 Laud Media

Put the pack in a folder called "hardware" and copy it here and check if it's good:

•					
🗸 🍃 Ce PC	SimpleModbusSlave/	ArduinoSLAVE3 Arduino 1.8.13			- 1
> 🔜 Bureau	Fichier Édition Croquis O	Outils Aide			
✓ I Documents		Formatage automatique	Ctrl+T		
🗸 📙 Arduino		Archiver le croquis			
✓ I hardware	SimpleModbusSlave	Reparer encodage & recharger	Ctrl+Mai+I		
ArduinoCore-sam-master	#include <	Moniteur série	Ctrl+Maj+M		
		Traceur série	Ctrl+Maj+L		
AllinyCore-master	/*	WiFi101 / WiFiNINA Firmware Upda	tor		
megaTinyCore-master	SimpleM			stion 3 6 s. 16	
> ightyCore-master		Type de carte: "Arduino Uno"	>	Gestionnaire de carte	
> Inicore-master	This ex	Récupérer les informations de la car	te	Arduno AVK Boards	rom the arduino
> 🧱 ArduinoCore-sam-master	Tt will			megaTinyCore (in sketchbook)	>gg of the led on
ATTinyCore-master	The male	Programmateur	>	MightyCore (in sketchbook)) in address 1 in
megaTinyCore-master	Ine val	Graver la sequence d'initialisation	·	MiniCore (in sketchbook)	ATmega328
	address	space namely hold.	ingRegs[]	+	ATmega168
MightyCore-master					ATmega88
MiniCore-master	In addit	ion to this the s	laves owr	n adc ch0 value y	ATmega48 cored in
> 📙 libraries	address	0 in its own addr	ess space	e holdingRegs[]	f or the m aster to
> 📙 Schneider Electric	be read.	The master will	use this	value to alter	the brightness of
۸					2

And the good progammer here and use it like this:

SimpleModbusSlave/	ArduinoSLAVE1 Arduino 1.8.13							
Fichier Édition Croquis C	utils Aide							
	Formatage automatique	Ctrl+T						
	Archiver le croquis							
SimpleModbusSlave	Réparer encodage & recharger	CH MILL						
noruringi	Gerer les bibliotneques	Ctrl+Maj+I						
memo=no1	Traceur série	Ctrl+Maj+L						
digitalW	WiFi101 / WiFiNINA Firmware Updater	r				Eichier Edition Croquis Outi	s Aido	
/* Note:	Type de carte: "ATmega328"	>	,		SimpleMod	husSlavoArduinoSLAVE1 Arduino 1	9 13	
The u	Clock: "External 16 MHz"	>	is not needed.	You co		BussiaveArduinoSEAVET Arduino I	.0.15	
size	BOD: "BOD 2.7V"	>	ng HOLDING REGS	SIZE 11	Fichier Édition C	Croquis Outils Aide		
boldi	Compiler LTO: "LTO disabled"	,	sing			Várifier/Compiler	Ctrl+R	
To	Variant: "328P / 328PA"	>	, sing.			Vermer/Compiler	Currix	
1.e.	Bootloader: "Yes (UART0)"	>	>			Téléverser	Ctrl+U	
norar	Port: "COM4 (Arduino Uno)"	>	(4)		SimpleMod	Téléverser avec un programmat	eur Ctrl+Maj+U	
analo	Récupérer les informations de la carte		(4);	1	hold	Exporter les binaires compilées	Ctrl+Alt+S	:
*/	Programmateur: "USBasp (MiniCore)"	>	STK500 as ISP (MiniCore)		110 1 01	Exporter les bindires complices	Carry acrossing 1	
	Graver la séquence d'initialisation		AVR ISP (MiniCore)		memo=	Afficher le dossier des croquis	Ctrl+K	
}			USBtinvISP (MiniCore)					
<			USBtinyISP slow (MiniCore)			Inclure une bibliotheque	>	
Téléversement terminé			 USBasp (MiniCore) 		digit	Ajouter un fichier		
avrdude: warning: cannot set sck per		USBasp slow (MiniCore)	k for i	argre	······			
avrdudet warning. cannot set sek per. A			Arduino as ISP (MiniCore)	ak for a	/* No	te:		
avidude: wa.		sck per.	Aumenice (AVR) (Minicore)	N TOL 1	, 10			

1-4- USBasp needed:

> 🍇 Contrôleurs de stockage > 📷 Contrôleurs IDE ATA/ATAPI > 🐗 Entrées et sorties audio > 🚍 Files d'attente à l'impression : 🕳 Lecteurs de disque

libusbK USB Devices Intel(R) Wireless Bluetooth(R)

> IP Périphériques de sécurité 🛺 Périphériques d'interface utilisateur

> Périphériques logiciels > ኪ Périphériques système Périphériques Universal Serial Bus

USBasp Ports (COM et LPT)

> 🕕 Souris et autres périphériques de pointage

>
Processeurs

> Microprogramme > Moniteurs > 💻 Ordinateur

> ÿ

>

>

When you connect for the first time the driver is not correctly installed So you need Zadig to reinstall the driver

🗾 Zadig \times Device Options Help Intel(R) Wireless Bluetooth(R) ~ Edit **More Information** Driver libusbK (v3.0.7.0) libusbK (v3.0.7.0) * WinUSB (libusb) USB ID 8087 0AAA libusb-win32 Reinstall Driver libusbK wcid 2 🗙 WinUSB (Microsoft) Driver Installation: SUCCESS Zadig 2.4.721 Now it's good: ✓ ▲ DESKTOP-BOUODF3 Appareils photo > > 🍃 Batteries > 🔚 Capteurs > 🏣 Cartes graphiques > 📕 Cartes hôte SD > 🖵 Cartes réseau > Claviers > 📑 Composants logiciels > 💐 Contrôleurs audio, vidéo et jeu > 🕴 Contrôleurs de bus USB



1-5- The programs for the both slaves:

The skectches are based on this example:

Fichier Édition Croquis	Outils Aide		
Nouveau	Ctrl+N		
Ouvrir	Ctrl+O		
Ouvert récemment	: >E	§	
Carnet de croquis	>		
Exemples	>	Ethernet	Progrator from add
Fermer	Ctrl+W	Firmata	>
Enregistrer	Ctrl+S	GSM	,then use this val
Enregistrer sous	Ctrl+Maj+S	LiquidCrystal	>
Mise en page	Ctrl+Maj+P	Robot Control	>ister (its own ad
Imprimer	Ctrl+P	Robot Motor	> huinhtara of
Drófórongog	Ctrl Wirgula	SD	, e brightness of
Freierences	Curt virgule	Servo	>
Quitter	Ctrl+Q	SpacebrewYun	>//////////////////////////////////////
#define LE	D 5	Stepper	>
#dotino ba	ud 1152	Temboo	>
	uu 1152	Retiré	>
#define ti	meout 2	Exemples pour ATmega328	
<pre>#define po</pre>	lling 1	AVR C code examples	>
#define re	try cou	EEPROM	>
		Optiboot flash read/write	>
		SoftwareSerial	>
// used to	toggle	SPI	, on the driver
#define Tx	EnableE	Wire	>
		Turana la substation la	
// mb - + - +	-1	Exemples depuis les bibliotneques personnalisees	the meater to at
<		SimpleModbusMaster	
		SimpleModbusSlave	SimpleModbusSlaveArduino
		SimpleModbusSlave DUE	>
		SimpleModbusSlaveSoftwareSerial	>
		SM	>
		∇	

The clone board: slave ID 1

#include <SimpleModbusSlave.h>

The arduino boards are slaves and there will be only one master: the HMI. This HMI will control leds on the both slave whenever you want.

I used the serial pin 0 and pin 1 (RX TX) and the pin 2 to control the link.

Serial settings: SERIAL_8N2: 1 start bit, 8 data bits, 2 stop bits and 115200 bauds according to those of the HMI.

The HMI lights ON/OFF the pin 5 of the slave 1 via holdingRegs[1] while the slave1 send the value 32765 via holdingRegs[0] to the HMI. A button on A0 of the arduino clone board light ON/OFF a lamp of the HMI via holdingRegs[2].

At the same time the HMI lights ON/OFF the pin 5 of the slave 3 via holdingRegs[1] while the slave 3 send the value 6666 via holdingRegs[0] to the HMI. A button on A0 of the arduino UNO board light ON/OFF a lamp of the HMI via holdingRegs[2].

SimpleModbusSlaveV10 supports function 3, 6 & 16.

This example code will receive the adc ch0 value from the arduino master. It will then use this value to adjust the brightness of the led on pin 9. The value received from the master will be stored in address 1 in its own address space namely holdingRegs[].

In addition to this the slaves own adc ch0 value will be stored in address 0 in its own address space holdingRegs[] for the master to be read. The master will use this value to alter the brightness of its own led connected to pin 9.

The modbus_update() method updates the holdingRegs register array and checks communication.

Note:

/*

The Arduino serial ring buffer is 64 bytes or 32 registers. Most of the time you will connect the arduino to a master via serial using a MAX485 or similar.

In a function 3 request the master will attempt to read from your slave and since 5 bytes is already used for ID, FUNCTION, NO OF BYTES and two BYTES CRC the master can only request 58 bytes or 29 registers.

In a function 16 request the master will attempt to write to your slave and since a 9 bytes is already used for ID, FUNCTION, ADDRESS, NO OF REGISTERS, NO OF BYTES and two BYTES CRC the master can only write

```
NO OF REGISTERS, NO OF BYTES and two BYTES CRC the master can only write
  54 bytes or 27 registers.
  Using a USB to Serial converter the maximum bytes you can send is
  limited to its internal buffer which differs between manufactures.
*/
#define LED 5
// Using the enum instruction allows for an easy method for adding and
// removing registers. Doing it this way saves you #defining the size
// of your slaves register array each time you want to add more registers
// and at a glimpse informs you of your slaves register layout.
enum
{
  // just add or remove registers and your good to go...
  // The first register starts at address 0
  ADC VAL, //adress 0
  PWM VAL, //ADDRESS 1
  BUTTON, // address 2
  HOLDING_REGS_SIZE // leave this one
  // total number of registers for function 3 and 16 share the same register array
  // i.e. the same address space
};
unsigned int holdingRegs[HOLDING REGS SIZE]; // function 3 and 16 register array
int memo;
void setup()
{
  /* parameters(HardwareSerial* SerialPort,
               long baudrate,
   unsigned char byteFormat,
              unsigned char ID,
               unsigned char transmit enable pin,
               unsigned int holding registers size,
               unsigned int* holding register array)
  */
 /* Valid modbus byte formats are:
   SERIAL 8N2: 1 start bit, 8 data bits, 2 stop bits
   SERIAL 8E1: 1 start bit, 8 data bits, 1 Even parity bit, 1 stop bit
   SERIAL 801: 1 start bit, 8 data bits, 1 Odd parity bit, 1 stop bit
   You can obviously use SERIAL 8N1 but this does not adhere to the
   Modbus specifications. That said, I have tested the SERIAL 8N1 option
   on various commercial masters and slaves that were suppose to adhere
   to this specification and was always able to communicate... Go figure.
   These byte formats are already defined in the Arduino global name space.
```

* /

```
modbus configure (& Serial, 115200, SERIAL 8N2, 1, 2, HOLDING REGS SIZE, holdingRegs);
  // modbus update comms(baud, byteFormat, id) is not needed but allows for easy update
  //of the port variables and slave id dynamically in any function.
 modbus update comms(115200, SERIAL 8N2, 1);
 pinMode(LED, OUTPUT);
  pinMode(14, INPUT);
}
void loop()
{
  // modbus update() is the only method used in loop(). It returns the total error
  // count since the slave started. You don't have to use it but it's useful
 // for fault finding by the modbus master.
 modbus update();
  //holdingRegs[ADC VAL] = analogRead(AO); // update data to be read by the master
                                            //to adjust the PWM
 holdingRegs[0] = 32565;
  //analogWrite(LED, holdingRegs[PWM VAL]>>2); // constrain adc value from the arduino
                                                //master to 255
 holdingRegs[2]=digitalRead(14);
 memo=holdingRegs[1];
 digitalWrite(LED, memo);
 /* Note:
    The use of the enum instruction is not needed. You could set a maximum allowable
    size for holdinRegs[] by defining HOLDING REGS SIZE using a constant and then access
    holdingRegs[] by "Index" addressing.
```

```
I.e.
holdingRegs[0] = analogRead(A0);
analogWrite(LED, holdingRegs[1]/4);
```

```
*/
```

}

The arduino board: slave ID 3

#include <SimpleModbusSlave.h>

/*

SimpleModbusSlaveV10 supports function 3, 6 & 16.

This example code will receive the adc ch0 value from the arduino master. It will then use this value to adjust the brightness of the led on pin 9. The value received from the master will be stored in address 1 in its own address space namely holdingRegs[].

In addition to this the slaves own adc ch0 value will be stored in address 0 in its own address space holdingRegs[] for the master to be read. The master will use this value to alter the brightness of its own led connected to pin 9. The modbus_update() method updates the holdingRegs register array and checks communication.

Note: The Arduino serial ring buffer is 64 bytes or 32 registers. Most of the time you will connect the arduino to a master via serial using a MAX485 or similar.

In a function 3 request the master will attempt to read from your slave and since 5 bytes is already used for ID, FUNCTION, NO OF BYTES and two BYTES CRC the master can only request 58 bytes or 29 registers.

In a function 16 request the master will attempt to write to your slave and since a 9 bytes is already used for ID, FUNCTION, ADDRESS, NO OF REGISTERS, NO OF BYTES and two BYTES CRC the master can only write 54 bytes or 27 registers.

Using a USB to Serial converter the maximum bytes you can send is limited to its internal buffer which differs between manufactures. */

#define LED 13

// Using the enum instruction allows for an easy method for adding and // removing registers. Doing it this way saves you #defining the size // of your slaves register array each time you want to add more registers // and at a glimpse informs you of your slaves register layout.

```
ADC_VAL, //adress 0
PWM_VAL, //ADDRESS 1
BUTTON, // address 2
HOLDING_REGS_SIZE // leave this one
// total number of registers for function 3 and 16 share the same register array
// i.e. the same address space
};
```

```
/* Valid modbus byte formats are:
    SERIAL 8N2: 1 start bit, 8 data bits, 2 stop bits
    SERIAL 8E1: 1 start bit, 8 data bits, 1 Even parity bit, 1 stop bit
    SERIAL 801: 1 start bit, 8 data bits, 1 Odd parity bit, 1 stop bit
    You can obviously use SERIAL 8N1 but this does not adhere to the
    Modbus specifications. That said, I have tested the SERIAL 8N1 option
    on various commercial masters and slaves that were suppose to adhere
    to this specification and was always able to communicate... Go figure.
    These byte formats are already defined in the Arduino global name space.
  */
  modbus configure (& Serial, 115200, SERIAL 8N2, 3, 2, HOLDING REGS SIZE, holdingRegs);
  // modbus update comms(baud, byteFormat, id) is not needed but allows for easy update
  //of the
  // port variables and slave id dynamically in any function.
  modbus update comms(115200, SERIAL 8N2, 3);
  pinMode(LED, OUTPUT);
   pinMode(14, INPUT);
}
void loop()
{
  // modbus update() is the only method used in loop(). It returns the total error
  // count since the slave started. You don't have to use it but it's useful
  // for fault finding by the modbus master.
 modbus update();
  //holdingRegs[ADC VAL] = analogRead(A0); // update data to be read by the master to
                                             //adjust the PWM
  holdingRegs[0] = 6666;
  //analogWrite(LED, holdingRegs[PWM VAL]>>2); // constrain adc value from the arduino
  //master to 255
  holdingRegs[2]=digitalRead(14);
 memo=holdingRegs[1];
  digitalWrite(LED, memo);
  /* Note:
     The use of the enum instruction is not needed. You could set a maximum allowable
     size for holdinRegs[] by defining HOLDING REGS SIZE using a constant and then access
     holdingRegs[] by "Index" addressing.
     T.e.
     holdingRegs[0] = analogRead(A0);
     analogWrite(LED, holdingRegs[1]/4);
  */
```

}

2-COOLMAY HMI and softwares needed:

Download here http://www.coolmay.com/Download-159-36-41.html

CoolMayHMI V5.906EN	2019-12-31	Down				
W-40B Editer setup	2018-04-08	J Down				
2-1-Install COOLMAY HMI soft and the very special driver : Install COOLMAY HMI and link the diplay like this :						

Special USB wire to the USB port of your PC -

- -The ethernet wire directly on the ethernet shield of your Arduino.
- Power on the display
- Have a look on the peripheral devices on your PC :
 - > 🕳 Lecteurs de disque
 - - HMI RNDIS
 - > Microprogramme
 - Monitours ς.

The device RNDIS appears as USB device, if not libusk use Zadig after restart in non signature driver mode. In the network list, there is only other networks.

(h.	SFR_AEEO Connecté, sécurisé Propriétés
	Déconnecter
¶.	SFR WiFi FON Ouvrir
(h.	SFR WiFi Mobile Sécurisé
97.	SFR_AEEO Ouvrir
Para Modi	mètres réseau & Internet fier des paramètres, pour rendre une connexion limitée.
<i>G</i> a Wi-Fi	ား Point d'accès Mode Avion sans fil mobile

It appears as HMI RNDIS, the wifi embedded functions are disabled. You need to install the driver in order to activate the wifi embeded on your display. Right click, Properties :



Update the driver :

Propriétés de : HMI RNDIS

 \times

IMI RNDIS Fournisseur du pilote : libusbK Date du pilote : 30/11/2014 Version du pilote : 30.7.0 Signataire numérique : USB\VID_045E&PID_0301 (libwdi autogenerated) Détails du pilote Affichez les détails concernant les fichiers du pilote installés. Mettre à jour le pilote Mettez à jour le pilote pour cet appareil. Restaurer le pilote Si le périphérique ne fonctionne pas après la mise à jour du pilote, réinstaller le pilote précédent. Désactiver l'appareil Désactivez l'appareil. Désinstaller l'appareil Désinstallez l'appareil du système (avancé).	Général	Pilote	Détails	Événements
Foumisseur du pilote : ibusbK Date du pilote : 30/11/2014 Version du pilote : 3.0.7.0 Signataire numérique : USB\VID_045E&PID_0301 (libwdi autogenerated) Détails du pilote Affichez les détails concernant les fichiers du pilote installés. Mettre à jour le pilote Mettez à jour le pilote pour cet appareil. Restaurer le pilote Si le périphérique ne fonctionne pas après la mise à jour du pilote, réinstaller le pilote précédent. Désactiver l'appareil Désinstallez l'appareil du système (avancé).	I	HMI RN	DIS	
Date du pilote :30/11/2014Version du pilote :3.0.7.0Signataire numérique :USB\VID_045E&PID_0301 (libwdi autogenerated)Détails du piloteAffichez les détails concernant les fichiers du pilote installés.Mettre à jour le piloteMettez à jour le pilote pour cet appareil.Restaurer le piloteSi le périphérique ne fonctionne pas après la mise à jour du pilote, réinstaller le pilote précédent.Désactiver l'appareilDésactivez l'appareil.Désinstaller l'appareilDésinstallez l'appareil du système (avancé).	Fourn	isseur du	pilote :	libusbK
Version du pilote :3.0.7.0Signataire numérique :USB\VID_045E&PID_0301 (libwdi autogenerated)Détails du piloteAffichez les détails concernant les fichiers du pilote installés.Mettre à jour le piloteMettez à jour le pilote pour cet appareil.Restaurer le piloteSi le périphérique ne fonctionne pas après la mise à jour du pilote, réinstaller le pilote précédent.Désactiver l'appareilDésactivez l'appareil.Désinstaller l'appareilDésinstallez l'appareil du système (avancé).OKAnnuler	Date	du pilote :		30/11/2014
Signataire numérique : USB\VID_045E&PID_0301 (libwdi autogenerated) Détails du pilote Affichez les détails concernant les fichiers du pilote installés. Mettre à jour le pilote Mettez à jour le pilote pour cet appareil. Restaurer le pilote Si le périphérique ne fonctionne pas après la mise à jour du pilote, réinstaller le pilote précédent. Désactiver l'appareil Désactivez l'appareil. Désinstaller l'appareil Désinstallez l'appareil du système (avancé).	Versio	on du pilot	te :	3.0.7.0
Détails du pilote Affichez les détails concernant les fichiers du pilote installés. Mettre à jour le pilote Mettez à jour le pilote pour cet appareil. Restaurer le pilote Si le périphérique ne fonctionne pas après la mise à jour du pilote, réinstaller le pilote précédent. Désactiver l'appareil Désactivez l'appareil. Désinstaller l'appareil Désinstallez l'appareil du système (avancé). OK Annuler	Signa	taire num	érique :	USB\VID_045E&PID_0301 (libwdi autogenerated)
Mettre à jour le pilote Mettez à jour le pilote pour cet appareil. Restaurer le pilote Si le périphérique ne fonctionne pas après la mise à jour du pilote, réinstaller le pilote précédent. Désactiver l'appareil Désactivez l'appareil. Désinstaller l'appareil Désinstallez l'appareil du système (avancé). OK Annuler	Dé	étails du p	ilote	Affichez les détails concernant les fichiers du pilote installés.
Restaurer le pilote Si le périphérique ne fonctionne pas après la mise à jour du pilote, réinstaller le pilote précédent. Désactiver l'appareil Désactivez l'appareil. Désinstaller l'appareil Désinstallez l'appareil du système (avancé). OK Annuler	Mett	re à jour le	e pilote	Mettez à jour le pilote pour cet appareil.
Désactiver l'appareil Désactivez l'appareil. Désinstaller l'appareil Désinstallez l'appareil du système (avancé). OK Annuler	Res	staurer le	pilote	Si le périphérique ne fonctionne pas après la mise à jour du pilote, réinstaller le pilote précédent.
Désinstaller l'appareil Désinstallez l'appareil du système (avancé). OK Annuler	Dés	activer l'ap	opareil	Désactivez l'appareil.
OK Annuler	Dési	nstaller l'a	ppareil	Désinstallez l'appareil du système (avancé).
				OK Annuler

Look for the driver in the PC :

Mettre à jour les pilotes - HMI RNDIS

Comment voulez-vous rechercher les pilotes ?

- → Rechercher automatiquement le logiciel de pilote à jour Windows va rechercher sur votre ordinateur et sur Internet le logiciel de pilote le plus récent pour votre appareil, sauf si vous avez désactivé cette fonctionnalité dans les paramètres d'installation de votre appareil.
- → Parcourir mon ordinateur à la recherche du logiciel de pilote Localisez et installez le logiciel de pilote manuellement.

Ar	nnuler
Go to the COOLMAY file you have just installed :	_
Mettre à jour les pilotes - HMI RNDIS	× AaBbCc AaBbCc. AaBbCcDa AaBbCcDa Rechercher un dossier ×
Rechercher des pilotes sur votre ordinateur	Sélectionnez le dossier qui contient les pilotes de votre matériel.
Rechercher les pilotes à cet emplacement : C\Kinco\Kinco HMIware v2.5\driver Parcourir ✓ Inclure les sous-dossiers	 Vidéos Disque local (C:) AdwCleaner FreeStyler Kinco Micrelec OA30_TOOL PerfLogs Program Files (x86) Adobe Artuino Artistic Licence Bonjour Common Files
Suivant Annul	CoolMay CoolMayHMI Examples library RunDat Crouzet Automatismes Fts3Pluraln CoolMayHMI OK Annuler

		\times
←	Mettre à jour les pilotes - HMI RNDIS	
	Rechercher des pilotes sur votre ordinateur	
	Rechercher les pilotes à cet emplacement :	
	C:\Program Files (x86)\CoolMay\CoolMayHMI Yarcourir	
	☑ Inclure les sous-dossiers	
	→ Choisir parmi une liste de pilotes disponibles sur mon ordinateur Cette liste affichera les pilotes disponibles compatibles avec l'appareil, ainsi que tous les pilotes dans la même catégorie que l'appareil.	
	Suivant Annuler	
Click	con choose in a list : CoolMayHMI	
←	Mettre à jour les pilotes - HMI RNDIS	
	Choisissez le pilote de périphérique à installer pour ce matériel.	
	Sélectionnez le fabricant et le modèle de votre périphérique matériel et cliquez sur Suiva vous avez un disque qui contient le pilote que vous voulez installer, cliquez sur Disque f	nt. Si ourni.
	Afficher les matériels compatibles	

 \times

Mo	odèle		
	HMI RNDIS		
	Périphérique série USB		
	CoolMayHMI		
Δ	Ce pilote n'a pas été signé numériquement !	[Disque fourni
	Pourquoi la signature du pilote est-elle importante ?		
		Suivent	Appular
		Sulvant	Annuler
Done :			

Mettre à jour les pilotes - CoolMayHMI #3

Windows a mis à jour vos pilotes

Windows a terminé l'installation des pilotes pour cet appareil :



CoolMayHMI

		Warman
Propriétés de : CoolMayHN	1I #3 ×	
Général Pilote Détails I	Événements	
CoolMayHMI #3		
Fournisseur du pilote :	CoolMay Corporation	
Date du pilote :	21/06/2006	
Version du pilote :	6.1.7600.16385	
Signataire numérique :	Non signé numériquement	
Détails du pilote	Affichez les détails concernant les fichiers du pilote installés.	
Mettre à jour le pilote	Mettez à jour le pilote pour cet appareil.	
Restaurer le pilote	Si le périphérique ne fonctionne pas après la mise à jour du pilote, réinstaller le pilote précédent.	
Désactiver l'appareil	Désactivez l'appareil.	
Désinstaller l'appareil	Désinstallez l'appareil du système (avancé).	
	Fermer Annuler	
It has been transform as a	network connector ·	

It has been transform as a network connector :

Fermer

击 Gestionnaire de périphériques	-	\times
Fichier Action Affichage ?		
🗸 📲 teclast		^
> 📮 Appareils mobiles		
> 🤪 Batteries		
> 😵 Bluetooth		
🗸 🔚 Capteurs		
E Bosch Accelerometer		
🌃 Microsoft Visual Studio Location Simulator Sensor		
> 🔙 Cartes graphiques		
> 🛄 Cartes hôte SD		
 Cartes réseau 		
Proadcom 802.11n Wireless SDIO Adapter		
CoolMayHMI #3		
WAN Miniport (IKEv2)		
WAN Miniport (L2TP)		
WAN Miniport (PTP)		
Claviers Contrôleurs audio vidéo et ieu		
Contrôleurs de bus LISB		
> 🖗 Contrôleurs de stockage		
Intrées et sorties audio		
> 🛱 Files d'attente à l'impression :		
> Tail Intel(R) Dynamic Platform and Thermal Framework		
> 🕳 Lecteurs de disque		
945 · · ·		÷

And a new network (the display's one) appears in the list as undentified network with no internet communication.



Now you're able to download the sketch you'll done with CoolmayHMI software.

2-2-IP adresses settings and MODBUS RTU communication:

You will have to work with 2 different addresses on your display :

- One is used to download the program
- The other is made for communicate with the ethernet peripheral device (Arduino , PLC....)

2-2-1 IP address for download :

You must use an IP address like 222.222.222.9 because the basic address of the display is 222.222.222.222

$\leftarrow \rightarrow \cdot \uparrow$	💘 « Réseau et Internet 🕨 Centre Réseau et partage	ڻ ~	Reche	ercher	Q
Fichier Edition Af	fichage Outils		- 📈	ĥĖ	🗙 🗸 🖃 🍯
Page d'accueil o configuration	lu panneau de Afficher les informations de base de votre ré connexions	éseau	et co	onfigure	er des
Modifier les carte	État de Ethernet 2 V Propriétés de Ethernet 3 X	d'acc	rès ·	Int	ernet
Modifier les partage avai	Gestion de réseau Partage	iexior	ns :	iW llte	-Fi (SFR_AEE0)
Options de multimédia	Propriétés de : Protocole Internet version 4 (TCP/IPv4)				
	Général Ce Les paramètres IP peuvent être déterminés automatiquement si votre réseau le permet. Sinon, vous devez demander les paramètres IP appropriés à votre administrateur réseau.	d'ace iexior	cès : ns :	Pas Ø Eth	s d'accès réseau nernet 3
	Obtenir une adresse IP automatiquement OUtiliser l'adresse IP suivante : Adresse IP : 222 . 222 . 222 . 9 Masque de sous-réseau : 255 . 255 . 0	résea tance	u ou VPI	N, ou conf	ìgurez un routeur ou
Voir aussi	Passerelle par défaut :	ou ac	cédez à	des infor	mations de
Options Inte Pare-feu Wir	Obtenir les adresses des serveurs DNS automatiquement OUtliser l'adresse de serveur DNS suivante : Serveur DNS préféré :				
) éléments	Serveur DNS auxiliaire :	-			ordinateur
	Valider les paramètres en quittant Avancé	٢			
	OK Annuler				

2-2-2 Launch as administrator :



And open the home made sketch IP search :



nodbus master serial\boutonRTU\boutonRTU.(Set OP PRM X Application(A) Debug(L) Individuation 0) - F Com. set Network set Alarm/Other Figure/Language Compile(C) F5 Ę Ы General Download(D) F6 2 HMI_Match Select Table HMI PRM: MT6070H (800*480 -1 (1 9 Set working PRMS(W) ... F7 Usb Disk Dat Permis. : Super □ Link2 Use • The good HMI here Batch modify(P)... ۲ F8 Linkl Set up Set keyboard PRMS(E) Fn. COM2 Device t Modbus RTU Slave Port: • • Modbus RTU slave Ľ Initial Acer() • Timeout: 200 ms 115200 Equipment 1 Rate: đ Clock Acer(O) The slave adress in • Dat Bits 8 b 💌 Stop bit:2 b CheckBit No • case of only one Ħ Sub Acer(S)... slave Attempts 8 \$ Fast Read 430 • + Data leng16 Target file path(H) ... Link2 Set up 8EN2 line 2 Save and backup(B)... Ŧ Device t Mitsubishi FX2N • COM2 21 Gallery(L) Timeout: 200 ms Ŧ Equipment0 1 Sound library(U) Dat Bits 7 b 💌 CheckBit Even -Stop bit:1 b -Product license(P) 8 + Fast Read D:0 Attempts 8 -Data leng • ICM(M) COM1 is RS232 Confirm(Y) plication Cancel(N) COM2 is RS485 : this one 115200 bauds like arduinos



Have a look here for multi slave communication:

1.2.2 Multi-machine Communication Settings

1) Open "Application --- Set Working Parameters --- Network Settings" .

Controller ID Address Mode: Select Extended Mode.

Extended communication ID switching interval: The default is 35ms, which can be adjusted according to actual communication.

Extended Mode Start ID: The default is 1, which is the first slave station number of the connected slave.

Each ID address register number: 100-30000 range can be set according to the actual register range setting of each slave.

The following figure shows: the HMI is connected with multiple slaves, the first slave station number is from 1. Number of each ID address register set 1000

When 4x0-4x999 indicates the address register of slave 0-999, 4x1000-4x1999 indicates 0-999 of slave 2. The register address, 4x2000-4x2999, represents register address 0-999 of slave 3... and so on.

Com. set Netw	ork set Aları	m/Other	Figure/Language	4
RS485/CAN	_Bus Multi	com.		
Cntrlr ID ad	ldr. mod Exter	- Ext	ded com. ID swit <mark>35</mark>	. . my
Extended mod	le intin <mark>1 - E</mark>	ldr.Eac	h ID addr. reg N 100	-
1993)				
Ethernet	settings		System time an	uto sy
Remote IP2:	222 . 222 . 222	2.222	☐ Auto sync function	
Remote IP3:	222 . 222 . 222	2 . 222	Syn. interval(Hou	2 🌲
Remote IP4:	222 . 222 . 222	2 . 222	From reg No. D: 20	00 🌲
			*Take 6 consecutive reg }	OH:MM:SS Y
Interacti	vo			
THUE ACU	vo	T 2 1 1 -		
Auto transf	ormation displa	Link I	reg No.D: 2100	
🗖 Report curr	ent pic. No. 🤇	Link 1 🔽	Reg No.D: 1180	

In case of only one slave:

Com set Notwork set	Alarm/Other Figure/Langua		Set OP PRM
General			Com. set Network set Alarm/Other Figure/Language
HMI PRM: MT607	7 0H (800*480 ▼ HMI Ma	atch Select Table	RS485/CAN_Bus Multi com.
Usb Disk Dat Permi	s.: Super 🔹 🗆 Link	2 Use	Cntrlr ID addr <mark>Standard -</mark> Extded com. ID swi 35 ms 🕇 t
			Ex.mode StartI 🚺 🝷 Each ID addr. reg 100 👻
Port: COM2 -	Device t Modbus RTU Sla	ve 👻	MultiHMI Share OFF
Rate: 115200 -	Timeout: 200 ms	Equipment1	Ethernet settings System Time Syn.
CheckBit No 🔹	Dat Bits 8 b 💌	Stop bit 2 b 🔹	Remote IP2: 222.222.222
Attempts 8	Fast Read 430	Data len <mark>s16</mark>	Remote IP3: 222.222.222 Syn. interval(Hq12
Link2 Set un	,		Remote IP4: 222.222.222 FromRegister 4x 200
Port: COM2 -	Device t <mark>Mitsubishi FX2</mark>	N 💌	Remote IP5: 222.222.222 *Take 6 consecutive reg HH: MM: SS Y
Rate: 9600 🔽	Timeout: 200 ms	Equipmen ¹⁰	Interactive
CheckBit Even 💌	Dat Bits 7 b 💌	Stop bit 1 b 💌	□ Auto transformation display s Link 1 - Register 4x 0
Attempts 8	Fast Read D:0	Data len <mark>0 </mark>	□ Report current pic. No. (OP-Link 1] Register 4x 0
	Confirm(Y)	lication Cancel(N)	Confirm(Y) plication Cancel(

2-3-An example of HMI:

The HMI looks like this: for nice and eyes catchy buttons have a look at my TUNNEL DE CHAUFFE instructable



<u>Controls for slave 1:</u> Slave 1 / holdingRegs[2] / register 40002

1:Window1			
	amp Attribute		×
•	Position Locked	Monitor reg Chann conn Link 1 -	Ctrl reg
	Left: 23 🌻	Elem typ: 4x • ?	□ Visibil cont
	Top: 39 📫	Register4x	Register0x: 8208
	Property	Show	
	Wide: 151 🌻	Type:Round Array nul 	♣ Array in 0 ♣:
	High: 152 🚔	Positive lo C Negative lo NoF1ash	ir - Font: 8X16 -
	OFF:	Status1: OFF Status2:	
	ON:	ON	
			Confirm(Y) Cancel(N)

Slave 1 / holdingRegs[1] / register 40001

	Bit operation button		×
▖▖▖▖▖▖▖▖ ᠈ः▙ः⊻▙▖▏₃॒ऽः⊙▕ध॓ゑ。 ᠈ۦ᠊▤•● ♥ ः♥ ः∎ ♥ ♀♀ੵ	Position Locked	Control Outline Text Set. *	Function
	Left 187 👤	Chann connLink 1 💌	C SetON C SetOFF
	Top: 14	Elem typ: 4x • ?	Keep type Alternatin Min.Press 0.0 Sec
	Property		
	High:251	Ctrl register	Protect
	Font:8X16	 Write addr.and addr. differen 	Grade Ordinary
~~	Aligr Midd 💌	WriteToRegister0x: 1	
			Confirm(Y) Cancel(N)

Slave 1 / holdingRegs[0] / register 40000

Digital display attribute		×	om	ponent I	a Langua	±{ ▼ -[f	English((U.S.A)]]	4	20	1
Position	Basic										8	
Locked S	Show forma Register 💌	Figure bl ² 📫										
Left: 459 👤 🔿	Chann conn Link 1 🔹	Pen secti ¹⁸ 🗘 h										
Top: 40 🗭 E	Elem typ: 4x • ?	Signed number										
Property	Register4x: 🚺 🔶	Backg Transpa		\bigcirc		>			>			
Wide: 332 🚖 🛙	Data type: 16 Bit 🔹	Off:										
High: 142 👤 🕻	Display digit 5 🚊	0n: 📃			/			/		/		
Backg	Decimals: 0											
	C	onfirm(<u>Y</u>) Cancel(<u>N</u>)										
	· · · · · · · · · · · · · · · · · · ·									::::	::::	

Controls for slave 3: Slave 3 / holdingRegs[2] / register 40202

	Lamp Attribute	• • • • • • • • • • • • • • • • • • •	×
	Position	Monitor reg	Ctrl reg
• • •	Left: 18 🌻	Elem typ: 4x • ?	□ Visibil cont
	Top: 294 👤	Register4x202	Register0x: 8208
	Property	Show	
	Wide: 147 🜲	Type:Round 💽 Array nul	Array in • :
	High: 150 🔶	● Positive lo ○ Negative lo NoF1ash	ir - Font: 8X16 -
	OFF:	Status1: OFF	
	ON:	ON	
			Confirm(Y) Cancel(N)

Slave 3 / holdingRegs[1] / register 40201

Bit operation button		X
Position	Control Outline Text Set. *	
Locked	Register	Function
Left 185 👤	Chann connLink 1 💌	C SetON C SetOFF
Top: 295	Elem typ: 4x 🔹 ?	Keep type Alternatin
Property	Register4x201	First Confirm
Wide:265	Ctrl register	Protect
High: <mark>145 🌻</mark>	□ Visibil cont	Password
Font:8X16 💌	□ Write addr.and addr. differen	Grade Ordinary 💌
AligrMidd 💌	WriteToRegister0x: 0	
		Confirm(Y) Cancel(N)

Slave 3 / holdingRegs[0] / register 40200

Digital display attribute		×	
Position	Basic		
Locked	Show forma Register 💌	Figure bl ² 🗦	
Left: 467 👤	Chann conn Link 1 -	Pen secti <mark>16 🌩</mark> h	
Top: 294 🚖	Elem typ: 4x • ?	Signed number	
Property	Register4x: 200	Backg Transpa	
Wide: 318 🜲	Data type: 16 Bit -	Off:	
High: 141 🜩	Display digit 5	0n: 🔼	
Backg	Decimals: 0		
	C	Confirm(Y) Cancel(N)	

2-4-Upload the sketch:

IOM	, Complie :			_		
Арр	lication(<u>A</u>)	Debug(<u>L</u>)	Individ	ы		
	Compile(<u>C</u>)	Í.	F5	'n		
2	Download(<u>D</u>)	F6			
	Set working	PRMS(W).	F7			
2	Batch modi	fy(<u>P</u>)	F8			
Б.	Set keyboar	d PRMS(<u>F</u>)				
ľ	Initial Acer(D				
٢	Clock Acer	0)				
#	Sub Acer(S)					
-	Target file p	ath(<u>H</u>)				
2	Save and ba	ackup(<u>B</u>)				
劉	Gallery(<u>L</u>)					
1	Sound libra	ry(<u>U</u>)				
6	Product lice	ense(P)				
a,	ICM(M)					
one	e and gene	rated in >	xx.hw6	i under c		
Cod	MayHM	I		\times		
Create file Success£; Size:54.422 KB						
			OK			

💺 📝 📜 =	Gérer	Windo	ows (C:)) ×
Fichier Accueil Partage Affichage	Outils de lecteur											~ ?
\leftarrow \rightarrow \checkmark \uparrow \clubsuit > Ce PC > Windows	(C:)		~	Ü	Q	Rech	nercher dans	: Windows	(C:)			
									- 🔏 🛛	è İ 🕻	×v	' 🖃 🌰
📰 Images		* ^	Non	n		^			Modifié	e le		Туре
📜 _simplemodbus master serial				-NII					12/21/2	2020 2.30	ΔΜ	Dossie
📙 _TP18 Domotique ZELIO3 COOLMAY				ntel					12/21/2	2020 5:26	AM	Dossie
boutonRTU				m.dat					12/20/2	2020 1:12	PM	Dossie
PROGRAMME COOLMAY				Morpho	(12/18/2	2020 6:09	AM	Dossie
			. I I	MSOCa	che				12/18/2	2020 1:54	AM	Dossie
OneDrive			📕 I	PerfLog	5				3/18/20	019 11:52	PM	Dossie
🗸 💄 Ce PC			I I	rogran	nData				12/21/2	2020 4:09	AM	Dossie
> 📃 Bureau			📜 I	Program	nmes				12/21/2	2020 5:15	AM	Dossie
> 🛅 Documents			📜 F	Program	nmes (x	x86)			12/20/2	2020 1:21	PM	Dossie
> 📰 Images			1	TestLog					12/18/2	2020 3:41	AM	Dossie
> h Musique			<u> </u>	Jtilisate	urs				12/17/2	2020 11:29	AM (Dossie
> Diets 3D			<u> </u>	NCH.CI	1				12/18/2	2020 1:54	AM	Dossie
Téléchargements				Nindov	/S				12/21/2	2020 4:52	AM	Dossie
Vidéos			** I	oouton	λτυ				12/21/2	2020 4:18	AM	Fichier
			~ (JUIBuilo	1				12/21/2	2020 4:18	AM	Fichier
Vindows (C:)				p.tmp					12/21/2	2020-10:37	2 AIVI	Fichier
🔉 🗳 Réseau												
		~	<									>
16 élément(s)												
16 éléments (espace libre : 65.5 Go)								11	2 Ko	🧢 Ordi	nateur	

Now you are able to download the sketch : if Download is not launch, restart CoolmayHMI Build as an Administrator.

	App	lication(<u>A</u>)	Debug(<u>L</u>)	Individu	a		
c		Compile(C)	F5			
1	*	Download	<u>(D)</u>	F6	2		
		Set workin	g PRMS(<u>W</u>)	F7	•		
100	۲	Batch mod	lify(<u>P</u>)	F8			
	Б	Set keyboa	rd PRMS(<u>F</u>)				
	ľ	Initial Acer	Û				
:	đ	Clock Acer(O)					
	#	Sub Acer(S)				
	-	Target file	path(<u>H</u>)				
1	2	Save and b	ackup(<u>B</u>)				
	劉	Gallery(<u>L</u>)					
ŀ	Ł	Sound libra	ary(<u>U)</u>				
	6	Product lic	ense(P)				
	a,	ICM(M)					
• -				/			

Right click :

GIUpdate V5.906				×	
OP HMI IP:	222.222.222.22	2	~	0	
OP HMI LR:	Local ~ Remo	te router sta	art po	pr 500	
OP TIM Up:	No ~ 🗆 Auto	o download	Sele	ct target file (H)	F2
			Net	work card information(P)	F1
Auto set IP, plea	ase wait	01/06/2020 1	Set r	network card IP (S)	
Fun Key			Ope WIN	n network connection(O) 10 ?????? (W)	
			Lang	guage selection(L)	>
GIUpdate V5.906	ated file :		×		
OP HMI IP: 222	2.222.222.222	~ 0			
OP HMI LR: Loc	al ~ Remote route	er start por 500)(
OP TIM Up: No	✓ Auto download	🗄 Down			
Filenam(C:\boutor	ıRTU.hw6				
Update success!	12/21/2	2020 10:32:31 AM			
elect the network w	here to download (create	d when driver update)	AND CHE	CK THE DEVICE.	X
GIUpdate V5.906				×	
OP HMI IP:	222.222.222.222	2	~	0	
OP HMI LR:	Local ~ Remot	e router star	t por	500	
OP TIM Up:	No ~ 🖂 Auto	download		Select target file (H)	E2
				Network and information	(D) E1
Update success	s!	12/21/2020 10:	34:28	Set network card IP (S)	(F) FI
N.				Open network connection WIN10 ?????? (W)	(0)
				Language selection(L)	

Select the wlan of the display :

🔶 🚽 🗸 📩 🦉 « Réseau et Internet 👂 Connevions rése	— L X A Recher
	eau > VU Rechercher dans : Connexions P
Fichier Edition Affichage Avancé Outils	E Sélecti
Organiser • Désactiver ce périphérique réseau Diag	nostiquer cette connexion »
Nom	Statut Nom du périphérique
C Ethernet 3	OD HMUD: 222 222 222 222
CoolMayHMI #3	
	OP HMI LR: Local ~ Remote router start por 500
	OP TIM Up: No V Auto download
	Filename C:\ESSAlmodbus.hw6
	Get Device Information 01/06/2020 15:31:23
2 élément(s) 1 élément sélectionné	
2 élément(s) 1 élément sélectionné CoolMayHMI #3	Drdinateur
2 élément(s) 1 élément sélectionné CoolMayHMI #3 Down :	Crdinateur
2 élément(s) 1 élément sélectionné CoolMayHMI #3 Down : GIUpdate V5.906	© Ordinateur
2 élément(s) 1 élément sélectionné CoolMayHMI #3 Down : GIUpdate V5.906 OP HMI IP: 222 222 2	© Ordinateur
2 élément(s) 1 élément sélectionné CoolMayHMI #3 Down : GIUpdate V5.906 OP HMI IP: 2222.222.2	© Ordinateur
2 élément(s) 1 élément sélectionné CoolMayHMI #3 Down : GIUpdate V5.906 OP HMI IP: 222.222.2 OP HMI LR: Local ~	<pre> Pordinateur Pordinateur X Pordinateur X Pordinateur X Pordinateur S Pordinateur</pre>
2 élément(s) 1 élément sélectionné CoolMayHMI #3 Down : GIUpdate V5.906 OP HMI IP: 222.222.2 OP HMI LR: Local ~ OP TIM Up: No ~	© Ordinateur 222.222 ↓ 0 Remote router start por 500 □ Auto download
2 élément(s) 1 élément sélectionné CoolMayHMI #3 Down : GIUpdate V5.906 OP HMI IP: 2222.222.2 OP HMI LR: Local ~ OP TIM Up: No ~	<pre> Provinateur /pre>
2 élément(s) 1 élément sélectionné CoolMayHMI #3 Down : GIUpdate V5.906 OP HMI IP: 2222.222.2 OP HMI LR: Local ~ OP TIM Up: No ~	<pre> Pordinateur /pre>
2 élément(s) 1 élément sélectionné CoolMayHMI #3 Down : GIUpdate V5.906 OP HMI IP: 2222.222.2 OP HMI LR: Local ~ OP TIM Up: No ~ Filenam C:\Psearch.hw6	<pre> Pordinateur ordinateur Pordinateur Pordinateur Pordinateur Pordinateur Pordinateur Pordinateur Pordinateur Pordinateur Pordinateur Pordinateur Pordinateur Por</pre>
2 élément(s) 1 élément sélectionné CoolMayHMI #3 Down : GIUpdate V5.906 OP HMI IP: 2222.222.2 OP HMI LR: Local ~ OP TIM Up: No ~ Filename C:VPsearch.hw6	<pre> Pordinateur ordinateur Pordinateur Pordinateur Pordinateur Pordinateur Pordinateur Pordinateur Pordinateur Pordinateur Pordinateur Pordinateur Pordinateur Pordinateur Pordina</pre>
2 élément(s) 1 élément sélectionné CoolMayHMI #3 Down : GIUpdate V5.906 OP HMI IP: 2222.222.2 OP HMI LR: Local ~ OP TIM Up: No ~ Filenam C:\Psearch.hw6	<pre> Pordinateur ordinateur Pordinateur Pordinateur Pordinateur Pordinateur Pordinateur Pordinateur Pordinateur Pordinateur Pordin</pre>

Success.

If the network doesn't run, stop and restart the master.

PART 2: HMI the SLAVE1, Clone the MASTER, UNO the SLAVE3

The schematic and the links between the devices remain the same. The Clone is now the master because you embed a program in state machine programming to control systems: you can't do it with an HMI as master because it doesn't respect safety machine rules.

The matrix of registers used for the communication:

Names of Packets received/sent	Address	Holding	Names
in the MASTER (enum array)	es in	registers	
	enum	addresses in	
	array	an array	
		defined by	
		TOTAL_NO_OF	
		_REGISTERS	
		(15 word max	
		for com)	
READ_SLAVE3_adr0_6666	0	10	holdingRegs[10]
			holdingRegs[_40011]
WRITE SLAVE3 adr1 LED13	1	1	holdingRegs[1]
			holdingRegs[WRITE_SLAVE3_adr1_LED13]
READ_SLAVE3_adr2_BUTTON	2	14	holdingRegs[14]
_40004	3	3	holdingRegs[3]
			holdingRegs[_40004]
_40005	4	4	holdingRegs[4]
			holdingRegs[_40005]
_40006	5	5	holdingRegs[5]
			holdingRegs[_40006]
_40007	6	6	holdingRegs[6]
			holdingRegs[_40007]
_40008	7	7	holdingRegs[7]
			holdingRegs[_40008]
_40009	8	8	holdingRegs[8]
			holdingRegs[_40009]
_40010	9	Not used	
_40011	10	10	holdingRegs[10]
			holdingRegs[_40011]
_40012	11	Not used	
_40013	12	Not used	
TOTAL_NO_OF_PACKETS	No		
	address		
	and		
	leave		
	this one		
		13	
		14	holdingRegs[14]
		15	

TOTAL_NO_OF_REGISTERS (here 15) must be equal or greater than TOTAL_NO_OF_PACKETS (here 12).

When you read a value from a slave you must do it via another register BECAUSE YOU REACH DIRECTLY the READ VALUES you want to proceed. For example, here I read a value '6666' from the slave3 and I want to display it on the HMI (slave1) on an LCD display with th address 40009. First: I store the value in holdingRegs[10] or holdingRegs[_40011] (but not holdingRegs[READ_SLAVE3_adr0_6666], not working). Second: I store holdingRegs[10] or holdingRegs[_40011] in holdingRegs[8] or holdingRegs[_40009].

2-1 The Clone as MASTER:

The skectch is based on this example:

SimpleMoo	busMas	terArduinoCLC	NE	Arduino 1.8.13		
Fichier Edition Nouveau	Croquis	Ctrl+N				
Ouvrir		Ctrl+O		_		
Ouvert ré	cemmen	t	> E §			
Carnet de	croquis		>			
Exemples			>	▲ Ethnoret		
Fermer		Ctrl+W		Einernei	1	o read a registe
Enregistre	er	Ctrl+S		Firmata	<i>.</i>	It will then us
Enregistre	er sous	Ctrl+Maj+S		GSIVI	(
Mice on r	200	Ctrl+Mai+D		LiquidCrystal Robert Control	(
Imprimor	age	Ctrl+D		Robot Control		ite a register (
imprimer		Cultr				ljusting the brig
Préférenc	es	Ctrl+Virgule		Sonia	(
Quitter		Ctrl+Q		SpacebrewYun	>	tion ////////////////////////////////////
#dofin	O LE	'D 5		Stepper	>	
#derin				Temboo	>	
#defin	e ba	ud 115.	2	Retiré	>	
#defin	e ti	meout .	2		0	
#defin	e po	lling	1	Exemples pour Armegasz	° , r	ate
#dofin	\circ ro	tru co		AVR C code examples	(
#derin	e re	cry_co	L	Optihe at flash read (write	(
				Coffware Carial	(
// use	d to	toggl		CDI	()	insmit pin on the
#defin	е Тх	Enable	F	SPI Wire	(
" dottii	C 11	шпарто	-	wire		
				Exemples depuis les biblio	thèc	
	+ -+			SimpleModbusMaster	>	SimpleModbusMasterArduino
				SimpleModbusMaster_DU	E >	
				SimpleModbusSlave	>	
				SimpleModbusSlave_DUE	>	
				SimpleModbusSlaveSoftw	ares	
				SM	>	
				∇		

The communication seems to be very slow. In order to get more speed:

-I use a state machine for a multitasking running
-I change some communication settings like this:
SERIAL_801: 1 start bit, 8 data bits, 1 Odd parity bit, 1 stop bit
115 200 bauds

115200 bauds

modbus_construct is a function which gives addresses and makes relationships between the registers of enum (coming from the slaves) and the local registers of the master:

```
//modbus_construct(&packets[THE REGISTER IN THE LIST], 1, READ_HOLDING_REGISTERS, register @,
//slave id, 0);
modbus_construct(&packets[READ_SLAVE3_adr0_6666], 3, READ_HOLDING_REGISTERS, 0, 1, 10);
modbus_construct(&packets[READ_SLAVE3_adr2_BUTTON], 3, READ_HOLDING_REGISTERS, 2, 1, 14);
modbus_construct(&packets[WRITE_SLAVE3_adr1_LED13], 3, PRESET_MULTIPLE_REGISTERS, 1, 1, 1);
modbus_construct(&packets[_40005], 1, READ_HOLDING_REGISTERS, 0x40005, 1, 4);
modbus_construct(&packets[_40004], 1, PRESET_MULTIPLE_REGISTERS, 0x40005, 1, 4);
modbus_construct(&packets[_40006], 1, PRESET_MULTIPLE_REGISTERS, 0x40006, 1, 5);
modbus_construct(&packets[_40008], 1, READ_HOLDING_REGISTERS, 0x40008, 1, 7);
modbus_construct(&packets[_40007], 1, PRESET_MULTIPLE_REGISTERS, 0x40007, 1, 6);
modbus_construct(&packets[_40009], 1, PRESET_MULTIPLE_REGISTERS, 0x40009, 1, 8);
```

Here the master operates and controls inputs and outputs registers:

```
//holdingRegs[ADC_VAL] = analogRead(A0); // update data to be read by the master
//to adjust the PWM
holdingRegs[_40006] = 32565;
holdingRegs[_40004] = digitalRead(14);
memo = holdingRegs[_40005];
digitalWrite(LED, memo);
//holdingRegs[_40009] = holdingRegs[10];
//holdingRegs[_40009] = holdingRegs[0]; NOT WORKING
holdingRegs[_40009] = holdingRegs[_40011];
//holdingRegs[_40009] = holdingRegs[READ_SLAVE3_adr0_6666];NOT WORKING
holdingRegs[WRITE_SLAVE3_adr1_LED13] = holdingRegs[_40008];
//holdingRegs[_40007] = holdingRegs[READ_SLAVE3_adr2_BUTTON]; NOT WORKING
holdingRegs[_40007] = holdingRegs[14];
```

If the network doesn't run, reset the master.

The sketch of the master:

```
SimpleModbusMasterArduinoCLONE
#include <SimpleModbusMaster.h>
#include <SM.h>
SM machine (&etape0);
/*
   The example will use packet1 to read a register from address 0 (the adc ch0 value)
   from the arduino slave (id=1). It will then use this value to adjust the brightness
   of an led on pin 9 using PWM.
   It will then use packet2 to write a register (its own adc ch0 value) to address 1
   on the arduino slave (id=1) adjusting the brightness of an led on pin 9 using PWM.
*/
#define LED 5
#define baud 115200
#define timeout 250
#define polling 100 // the scan rate
#define retry_count 5 // 10
// used to toggle the receive/transmit pin on the driver
#define TxEnablePin 2
// The total amount of available memory on the master to store data OF ENUM ARRAY
#define TOTAL NO OF REGISTERS 15
// This is the easiest way to create new packets
// Add as many as you want.
// TOTAL NO OF PACKETS is automatically updated.
```

```
enum
ł
  // just add or remove registers and your good to go...
  // The first register starts at address 0
  READ_SLAVE3_adr0_6666, //address 0 6666
  WRITE_SLAVE3_adr1_LED13, //address 1 LED 13
  READ SLAVE3 adr2 BUTTON, // address 2 BUTTON AO
  _40004, // address 3
  _40005, // address 4
  _40006, // address 5
  _40007, // address 6
  _40008, // address 7
  _40009, // address 8
  _40010, // address 9
  _40011, // address 10
  _40012, // address 11
   40013, // address 12
  TOTAL_NO_OF_PACKETS // leave this one
  // total number of registers for function 3 and 16 share the same register array
  // i.e. the same address space
};
// Create an array of Packets to be configured
Packet packets[TOTAL NO OF PACKETS];
// Masters register array
unsigned int holdingRegs[TOTAL NO OF REGISTERS]; // function 3 and 16 register array
int memo;
int memo2;
void setup()
{
  /* parameters(HardwareSerial* SerialPort,
                long baudrate,
    unsigned char byteFormat,
                unsigned char ID,
                unsigned char transmit enable pin,
                unsigned int holding registers size,
                unsigned int* holding register array)
  */
  /* Valid modbus byte formats are:
     SERIAL 8N2: 1 start bit, 8 data bits, 2 stop bits
     SERIAL_8E1: 1 start bit, 8 data bits, 1 Even parity bit, 1 stop bit
     SERIAL 801: 1 start bit, 8 data bits, 1 Odd parity bit, 1 stop bit
     You can obviously use SERIAL 8N1 but this does not adhere to the
     Modbus specifications. That said, I have tested the SERIAL 8N1 option
     on various commercial masters and slaves that were suppose to adhere
```

```
to this specification and was always able to communicate... Go figure.
    These byte formats are already defined in the Arduino global name space.
 * /
 // Initialize each packet
 //modbus construct(&packets[THE REGISTER IN THE LIST], 1, READ HOLDING REGISTERS, register @,
 //slave id, 0);
 modbus construct (&packets [READ SLAVE3 adr0 6666], 3, READ HOLDING REGISTERS, 0, 1, 10);
 modbus construct (&packets [READ SLAVE3 adr2 BUTTON], 3, READ HOLDING REGISTERS, 2, 1, 14);
 modbus_construct(&packets[WRITE_SLAVE3_adr1_LED13], 3, PRESET_MULTIPLE_REGISTERS, 1, 1, 1);
 modbus_construct(&packets[_40005], 1, READ_HOLDING_REGISTERS, 0x40005, 1, 4);
 modbus_construct(&packets[_40004], 1, PRESET_MULTIPLE_REGISTERS, 0x40004, 1, 3);
 modbus_construct(&packets[_40006], 1, PRESET_MULTIPLE_REGISTERS, 0x40006, 1, 5);
 modbus_construct(&packets[_40008], 1, READ_HOLDING_REGISTERS, 0x40008, 1, 7);
 modbus_construct(&packets[_40007], 1, PRESET_MULTIPLE_REGISTERS, 0x40007, 1, 6);
 modbus_construct(&packets[_40009], 1, PRESET_MULTIPLE_REGISTERS, 0x40009, 1, 8);
 //modbus construct(&packets[holdingRegs[0]], 1, PRESET MULTIPLE REGISTERS, 1, 1, 0);
 // Initialize the Modbus Finite State Machine
 modbus_configure(&Serial, baud, SERIAL_801, timeout, polling, retry_count, TxEnablePin, packets,
                                                                      TOTAL NO OF PACKETS, holdingRegs);
 pinMode(LED, OUTPUT);
 pinMode(14, INPUT); //AO
void loop()
{
 EXEC (machine);
 // modbus update() is the only method used in loop(). It returns the total error
 // count since the slave started. You don't have to use it but it's useful
 // for fault finding by the modbus master.
 modbus update();
State etape0() {
 //holdingRegs[ADC VAL] = analogRead(A0); // update data to be read by the master
 //to adjust the PWM
 holdingRegs[ 40006] = 32565;
 holdingRegs[ 40004] = digitalRead(14);
 memo = holdingRegs[ 40005];
 digitalWrite(LED, memo);
 //holdingRegs[ 40009] = holdingRegs[10];
 //holdingRegs[ 40009] = holdingRegs[0]; NOT WORKING
 holdingRegs[ 40009] = holdingRegs[ 40011];
 //holdingRegs[ 40009] = holdingRegs[READ SLAVE3 adr0 6666];NOT WORKING
 holdingRegs[WRITE SLAVE3 adr1 LED13] = holdingRegs[ 40008];
 //holdingRegs[ 40007] = holdingRegs[READ SLAVE3 adr2 BUTTON]; NOT WORKING
 holdingRegs[ 40007] = holdingRegs[14];
 /* Note:
    The use of the enum instruction is not needed. You could set a maximum allowable
    size for holdinRegs[] by defining HOLDING REGS SIZE using a constant and then access
    holdingRegs[] by "Index" addressing.
    holdingRegs[0] = analogRead(A0);
    analogWrite(LED, holdingRegs[1]/4);
 * /
```

}

}

2-2 The UNO as SLAVE3:

The skectch is based on this example:

🥯 SimpleModbusSlaveArduinoSLAVE3 | Arduino 1.8.13

Fichi	er Édition Croquis	Outils Aide		
	Nouveau	Ctrl+N		
	Ouvrir	Ctrl+O		
	Ouvert récemment	>	5	
	Carnet de croquis	>	odbugglarro h	
	Exemples	>	▲ Ethernet	>
	Fermer	Ctrl+W	Eirmata	>
	Enregistrer	Ctrl+S	GSM	>
	Enregistrer sous	Ctrl+Maj+S	LiquidCrystal	>
	Mise en page	Ctrl+Maj+P	Robot Control	>
	Imprimer	Ctrl+P	Robot Motor	>
	Defference	Childhamla	SD	>
	Preferences	Ctri+virguie	Servo	, function 3, 6 &
	Quitter	Ctrl+Q	SpacebrewYun	>
	This exa	ample d	Stepper	>e the adc ch0 va
	Tt will	then u	Temboo	'b adjust the bri
	The rela		Retiré	> adjust the sti
	Ine val	ue rece	Exemples pour ATmega328	aster will be st
	address	space	AVR C code examples	,egs[].
			EEPROM	>
	In addi	tion to	Optiboot flash read/write	's own adc ch0 va
	addrogg	0 in i	SoftwareSerial	>mage heldingPog
	auuress		SPI	>ppace norungkey
	be read	. The m	Wire	,this value to al
Télé	eversement terminé		Exemples depuis les bibliot	nèc
T.O	croquis	ntilis	SimpleModbusMaster	> 1%) de l'espace
цС т	CIOQUID 	uciiio	SimpleModbusMaster_DUE	\rightarrow
ье	s variabl	les gio	SimpleModbusSlave	> SimpleModbusSlaveArduino
			SimpleModbusSlave_DUE	>
<			SimpleModbusSlaveSoftwa	rð
1			SM	>
	O Taparici n	our rachard	∇	

The communication seems to be very slow. In order to get more speed:

-I use a state machine for a multitasking running

-I change some communication settings like this:

SERIAL_801: 1 start bit, 8 data bits, 1 Odd parity bit, 1 stop bit 115200 bauds

SimpleModbusSlaveArduinoSLAVE3

```
#include <SimpleModbusSlave.h>
#include <SM.h>
```

```
SM VALEUR(&etape0);
SM MACHINE(&etape10);
/*
```

SimpleModbusSlaveV10 supports function 3, 6 & 16.

This example code will receive the adc ch0 value from the arduino master. It will then use this value to adjust the brightness of the led on pin 9. The value received from the master will be stored in address 1 in its own address space namely holdingRegs[].

In addition to this the slaves own adc ch0 value will be stored in address 0 in its own address space holdingRegs[] for the master to

be read. The master will use this value to alter the brightness of its own led connected to pin 9. The modbus update() method updates the holdingRegs register array and checks Note: The Arduino serial ring buffer is 64 bytes or 32 registers. Most of the time you will connect the arduino to a master via serial using a MAX485 or similar. In a function 3 request the master will attempt to read from your slave and since 5 bytes is already used for ID, FUNCTION, NO OF BYTES and two BYTES CRC the master can only request 58 bytes or 29 registers. In a function 16 request the master will attempt to write to your slave and since a 9 bytes is already used for ID, FUNCTION, ADDRESS, NO OF REGISTERS, NO OF BYTES and two BYTES CRC the master can only write 54 bytes or 27 registers. Using a USB to Serial converter the maximum bytes you can send is limited to its internal buffer which differs between manufactures.

*/

#define LED 13

// Using the enum instruction allows for an easy method for adding and // removing registers. Doing it this way saves you #defining the size // of your slaves register array each time you want to add more registers // and at a glimpse informs you of your slaves register layout.

enum {

// just add or remove registers and your good to go... // The first register starts at address 0 ADC_VAL, //adress 0 PWM_VAL, //ADDRESS 1 BUTTON, // address 2 HOLDING_REGS_SIZE // leave this one // total number of registers for function 3 and 16 share the same register array // i.e. the same address space };

```
unsigned int holdingRegs[HOLDING REGS SIZE]; // function 3 and 16 register array
int memo;
int valeur;
```

{

}

{

}

```
void setup()
  /* parameters(HardwareSerial* SerialPort,
                long baudrate,
    unsigned char byteFormat,
                unsigned char ID,
                unsigned char transmit enable pin,
                unsigned int holding registers size,
                unsigned int* holding register array)
  */
 /* Valid modbus byte formats are:
    SERIAL 8N2: 1 start bit, 8 data bits, 2 stop bits
    SERIAL 8E1: 1 start bit, 8 data bits, 1 Even parity bit, 1 stop bit
    SERIAL 801: 1 start bit, 8 data bits, 1 Odd parity bit, 1 stop bit
    You can obviously use SERIAL 8N1 but this does not adhere to the
    Modbus specifications. That said, I have tested the SERIAL 8N1 option
    on various commercial masters and slaves that were suppose to adhere
    to this specification and was always able to communicate... Go figure.
    These byte formats are already defined in the Arduino global name space.
  */
 modbus configure (& Serial, 115200, SERIAL 801, 3, 2, HOLDING REGS SIZE, holdingRegs);
  // modbus_update_comms(baud, byteFormat, id) is not needed but allows for easy update
  //of the
  // port variables and slave id dynamically in any function.
 modbus update comms(115200, SERIAL 801, 3);
 pinMode(LED, OUTPUT);
 pinMode(14, INPUT);
void loop()
  EXEC (VALEUR);
  EXEC (MACHINE);
  // modbus_update() is the only method used in loop(). It returns the total error
  // count since the slave started. You don't have to use it but it's useful
  // for fault finding by the modbus master.
 modbus_update();
```

```
State etape10() {
 //holdingRegs[ADC VAL] = analogRead(A0); // update data to be read by the master to
                                             //adjust the PWM
 holdingRegs[0] = valeur;
  //analogWrite(LED, holdingRegs[PWM VAL]>>2); // constrain adc value from the arduino
  //master to 255
 holdingRegs[2]=digitalRead(14);
 memo=holdingRegs[1];
 digitalWrite(LED,memo);
  /* Note:
     The use of the enum instruction is not needed. You could set a maximum allowable
     size for holdinRegs[] by defining HOLDING REGS SIZE using a constant and then access
     holdingRegs[] by "Index" addressing.
     I.e.
     holdingRegs[0] = analogRead(A0);
     analogWrite(LED, holdingRegs[1]/4);
  */
}
State etape0() {
  valeur=6666;
  if (VALEUR.Timeout(200)) VALEUR.Set(etape1);
}
State etape1() {
  valeur=2222;
  if (VALEUR.Timeout(200)) VALEUR.Set(etape0);
```

2-3 The HMI as SLAVE1:

An example of a dashboard:

}



Communication settings:

	1			Juio			
HMI PRM:	MT	6070I	H (800*4	80 -	HMI Match	Select Ta	ıb1
Usb Disk	Dat Per	rmis.	: Super	•	Link2 U	se	
Link1 9	Set up-						
Port:	COM2	• I	Device t	odbus R	TU Master		•
Rate:	115200	•	Fimeout:2	00 ms	Equ	ipmen 1	
CheckBit	Odd	• I	Dat Bits	b 🔻	Sto	p bit 1 b	•
Attempts	8	•	Fast Read	43 <mark>0</mark>	🔹 Dat	a 1enş <mark>16</mark>	•
Link2 S	Set up-						
Port:	COM1	- I	Device t	odbus F	TU Master		•
Rate:	9600	-	limeout:2	00 ms	Equ	ipmen 0	
CheckBit	No	• I	Dat Bits 8	b 🔻	Sto	p bit 1 b	
Attempts	8	➡ E	ast Read	43 <mark>0</mark>	🗧 Dat	a leng <mark>0</mark>	•
				Confir	m(Y) lic	ation	Ce
						<u> </u>	
P PRM							
om. set N	letwork se	t Ala	arm/Other	Figure	/Language		
om. set N RS485/(letwork se CAN_Bus	et Ala Mul	arm/Other ti com.	Figure	/Language		
om.set N RS485/(Cntrlr	letwork se C <mark>AN_Bus</mark> ID addrS	t Al Mul Standa	arm/Other ticom. ard -	Figure Cxtded	/Language	swi <mark>35 ms</mark>	
om. set N RS485/(Cntrlr Ex. mode	letwork se CAN_Bus ID addrS StartI1	et Al Mul	arm/Other ti com. ard - H	Figure Extded	/Language com. ID :) addr. re	_{swi} 35 ms eg 100	•
om.set N RS485/(Cntrlr Ex.mode MultiHM	letwork se CAN_Bus ID addrS StartI I ShareC	et Ala Mul Standa	arm/Other ti com. ard - H ti com. Ard - H t	Figure Extded	/Language com. ID ;) addr. re	swi <mark>35 ms</mark> eg 100	-
om. set N RS485/(Cntrlr Ex. mode MultiHM Etherne	letwork se CAN_Bus ID addr StartI I Share t sett	et Ali Mul Standa)FF ings	arm/Other ti com. ard • H • H • H	Figure Extded Each II	/Language com. ID :) addr. ro stem Ti	swi <mark>35 ms</mark> eg 100 ne Syn. -	
om. set N RS485/(Cntrlr Ex. mode MultiHM Etherne Remote	letwork se CAN_Bus ID addr StartI I Share t sett IP2: 222	et Ala Mul Standa DFF ings . 222	arm/Other ti com. ard • F • F • F • F • F • F • F • F	Figure Extded Each II	/Language com. ID ;) addr. ro stem Tin Auto sync fu	swi <mark>35 ms</mark> eg 100 ne Syn. – nction	×
om. set N RS485/(Cntrlr Ex. mode MultiHM Etherne Remote Remote	letwork se CAN_Bus ID addr StartI I Share t sett IP2: 222 IP3: 222	et Ala Mul Standa DFF ings . 222 . 222	arm/Other ti com. ard • F • F • F • C • C • C • C • C • C • C • C	Figure Extded Each II Syn	/Language com. ID ;) addr. ro stem Tin Auto sync fu a. interva	swi <mark>35 ms</mark> eg 100 ne Syn. – nction al (Hq 12	
om. set N RS485/(Cntrlr Ex. mode MultiHM Etherne Remote Remote Remote	letwork se CAN_Bus ID addr StartI I Share et sett IP2: 222 IP3: 222 IP4: 222	et Ali Mu1 Standa DFF ings . 222 . 222 . 222	arm/Other ti com. ard • F • F • F • F • F • F • F • F	Figure Extded Each II Syn Fro	/Language com. ID ;) addr. rd stem Tin Auto sync fu n. interva	swi <mark>35 ms</mark> eg 100 ne Syn. – nction al (Hq 12 r 4x 200	
om. set N RS485/(Cntrlr Ex. mode MultiHM Etherne Remote Remote Remote Remote	letwork se CAN_Bus ID addr[S StartI] I Share t sett IP2: 222 IP3: 222 IP3: 222 IP4: 222 IP5: 222	et Ali Mu1 Standa DFF ings . 222 . 222 . 222 . 222	arm/Other ti com. ard • F • F • F • F • F • F • F • F	Figure Extded Each II Syn Fro *Tal	/Language com. ID :) addr. ro stem Tin Auto sync fu h. interva omRegiste	swi 35 ms eg 100 ne Syn nction al (Hq12 r 4x 200 ive reg HH: M	

□ Report current pic. No. (OP-Link 1 - Register4x 0

 $Confirm(\underline{Y})$ plication Cancel(N)

Supervision settings for the MASTER:

🚅 1:Window1	Lamp Attribute		×
	Position Locked Left: 23 🔹 Top: 39 🔹	Monitor reg Chann connLink 1 • Elem typ: 4x • ? Register4x	Ctrl reg □ Visibil cont Register0x: 8208 →
	Property Wide: 151 🔹 High: 152 🜲 OFF:	Show Type:Round • Array nul • Positive lo C Negative le NoFlashi Status1: OFF Status2: ON	 Array in 0 ↓: r ▼ Font: 8X16 ▼
			Confirm(Y) Cancel(N)
	Bit operation but Position Locked Left 187 Top: 18 Property Wide: 257 High: 251 Font: 8X16 Aligr Midd	Control Outline Text Set. * Register Chann connLink 1 • Chann connLink 1 • E1em typ: 4x • ? Register4x Ctrl register Visibil cont Write addr.and addr. differen WriteToRegister0x: 1 •	Function C SetON C SetOFF Keep type C Alternatin Min.Press 0.0 Sec First Confirm Protect Password GradeOrdinary
			Confirm(Y) Cancel(N)

1:Window1		
Digital display attribute	×	
Position Basic		
□ Locked Show forma Registe	er ▼ Figure b1 ² ↓	
Left: 456 Chann conn Link 1	▼ Pen secti18 ♣h	
Top: 58 🗧 Elem typ: 4x 🔹	? 🗆 Signed number	
Property Register4x:	Backg Transpa	
Wide: 332 🔿 Data type: 16 Bit	▪ Off: I	
High: 142 Display digit 5	🝨 0n: 🗾	-
Backg		
	Confirm(Y) Cancel(N)	

Supervision settings for the SLAVE3:

	Lamp Attribute		×
	Position	Monitor reg	Ctrl reg
	Left: 18	Elem typ: 4x • ?	□ Visibil cont
	Top: 294 👤	Register4x	Register0x: 8208
•	Property	Show]
	Wide: 147 🌩	Type:Round 🔹 Array nul	Array in 0 <table-cell-rows>:</table-cell-rows>
	High: 150	Positive Io: Negative Iv NoF1ash	nir ▼ Font: 8X16 ▼
	OFF:	Status1: OFF Status2:	
	ON:	ON	
			Confirm(<u>Y</u>) Cancel(<u>N</u>)

	Bit operation button		×
	Position	Control Outline Text Set. *	
	Locked	Register	Function
	Left 185 👤	Chann connLink 1 🗸	C SetON C SetOFF
	Top: 295	Elem typ: 4x 🔹 ?	Keep type Alternatin
		Register4x	Min.Press 0.0 Sec
	Property		
	Wide 200	Ctrl register	Protect
	High: 145	□ Visibil cont	Password
	Font:8X16 -	C Write addr.and addr. differen	Grade Ordinary 💌
	Aligr Midd 🔹	WriteToRegister0x: 0	
			Confirm(<u>Y</u>) Cancel(<u>N</u>)
Digital display attribute		×	
Position Basic			
Locked Show forma Regis	ster 🗾 Figure	b]2	
Top: 301 Chann conn Link	▼ Pen sec	tinumber	
Property Register4x: 9	Backg	Transpa	
Wide: 318 🜩 Data type: 16 Bi	t 🔹 Off:		
High: 141 🔶 Display digit 5	🗐 0n: 📘		
Backg	Ţ		
	Confirm(Y)	Cancel(N)	