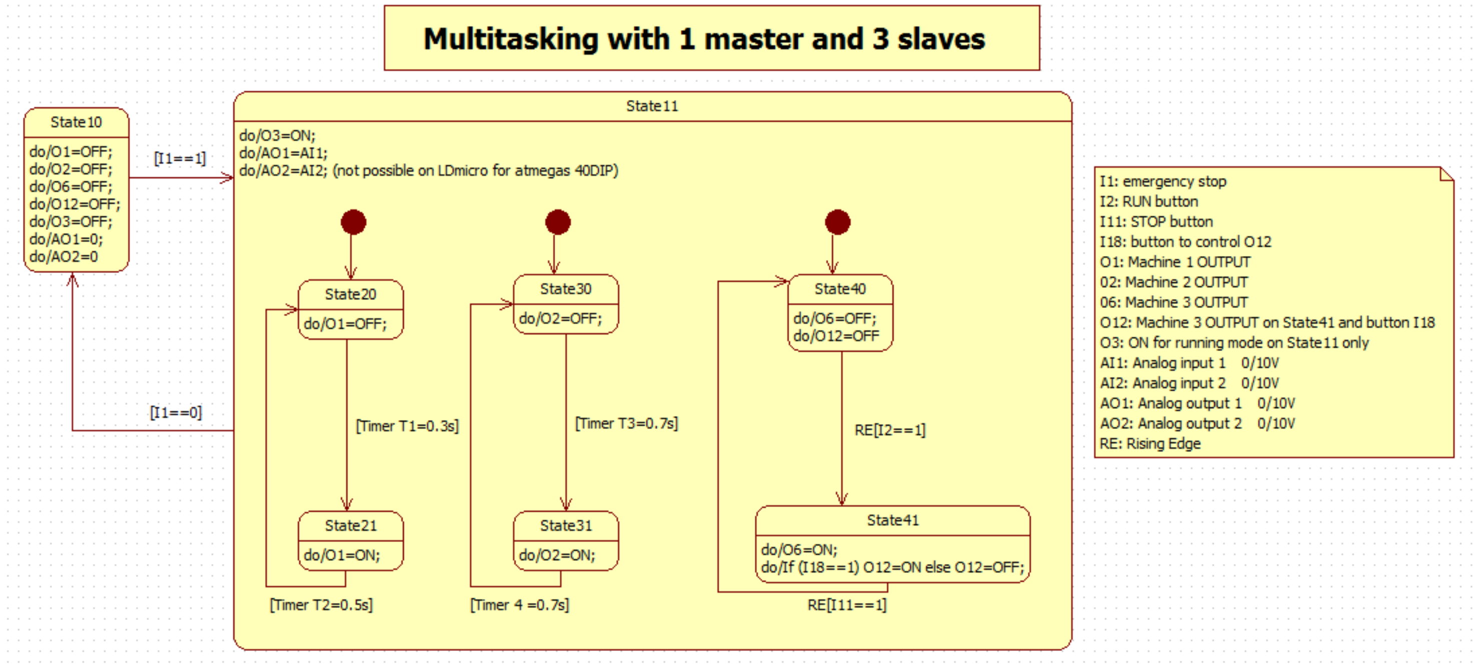
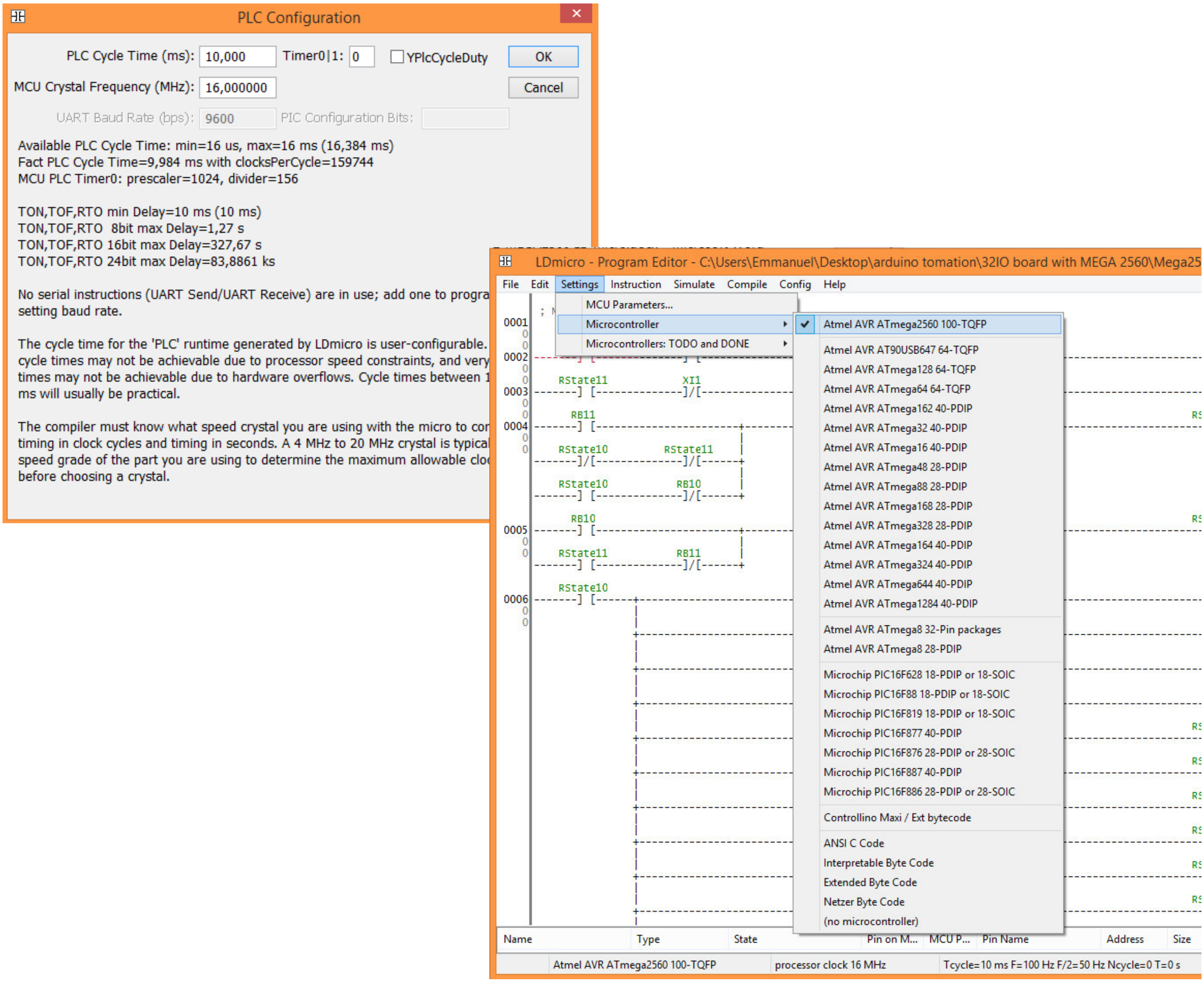
1-Multitasking with ladder programming on LDmicro4.2 for MEGA 2560 via USB

1-0 The machines to program:



1-1 Settings on LDmicro:



PLC Configuration
OK
Cancel
LDmicro - Program Editor - C:\Users\Emmanuel\Desktop\arduino tomation\32IO board with MEGA 2560\Mega25
File
Edit
Settings
Instruction
Simulate
Compile
Config
Help
MCU Parameters...
Microcontroller
Microcontrollers: TODO and DONE
Atmel AVR AT90USB647 64-TQFP
Atmel AVR ATmega128 64-TQFP
Atmel AVR ATmega64 64-TQFP
Atmel AVR ATmega162 40-PDIP
Atmel AVR ATmega32 40-PDIP
Atmel AVR ATmega16 40-PDIP
Atmel AVR ATmega48 28-PDIP
Atmel AVR ATmega88 28-PDIP
Atmel AVR ATmega168 28-PDIP
Atmel AVR ATmega328 28-PDIP
Atmel AVR ATmega164 40-PDIP
Atmel AVR ATmega324 40-PDIP
Atmel AVR ATmega644 40-PDIP
Atmel AVR ATmega1284 40-PDIP
Atmel AVR ATmega8 32-Pin packages
Atmel AVR ATmega8 28-PDIP
Microchip PIC16F628 18-PDIP or 18-SOIC
Microchip PIC16F88 18-PDIP or 18-SOIC
Microchip PIC16F819 18-PDIP or 18-SOIC
Microchip PIC16F877 40-PDIP
Microchip PIC16F876 28-PDIP or 28-SOIC
Microchip PIC16F887 40-PDIP
Microchip PIC16F886 28-PDIP or 28-SOIC
Controllino Maxi / Ext bytecode
ANSI C Code
Interpretable Byte Code
Extended Byte Code
Netzer Byte Code
(no microcontroller)
0001
0002
0003
0004
0005
0006
RState11
XI1
RB11
RState10
RState11
RState10
RB10
RB10
RState11
RB11
RState10
Name
Type
State
Pin on M...
MCU P...
Pin Name
Address
Size

PLC Cycle Time (ms): 10,000 Timer0|1: 0 □YPlcCycleDuty

MCU Crystal Frequency (MHz): 16,000000

UART Baud Rate (bps): 9600 PIC Configuration Bits:

Available PLC Cycle Time: min=16 us, max=16 ms (16,384 ms)
Fact PLC Cycle Time=9,984 ms with clocksPerCycle=159744
MCU PLC Timer0: prescaler=1024, divider=156

TON,TOF,RTO min Delay=10 ms (10 ms)
TON,TOF,RTO 8bit max Delay=1,27 s
TON,TOF,RTO 16bit max Delay=327,67 s
TON,TOF,RTO 24bit max Delay=83,8861 ks

No serial instructions (UART Send/UART Receive) are in use; add one to progra
setting baud rate.

The cycle time for the 'PLC' runtime generated by LDmicro is user-configurable.
cycle times may not be achievable due to processor speed constraints, and very
times may not be achievable due to hardware overflows. Cycle times between 1
ms will usually be practical.

The compiler must know what speed crystal you are using with the micro to cor
timing in clock cycles and timing in seconds. A 4 MHz to 20 MHz crystal is typical
speed grade of the part you are using to determine the maximum allowable clo
before choosing a crystal.

☑ Atmel AVR ATmega2560 100-TQFP

Atmel AVR ATmega2560 100-TQFP processor clock 16 MHz Tcycle=10 ms F=100 Hz F/2=50 Hz Ncycle=0 T=0 s

1-2-Program the ladder: a translation of the state diagram

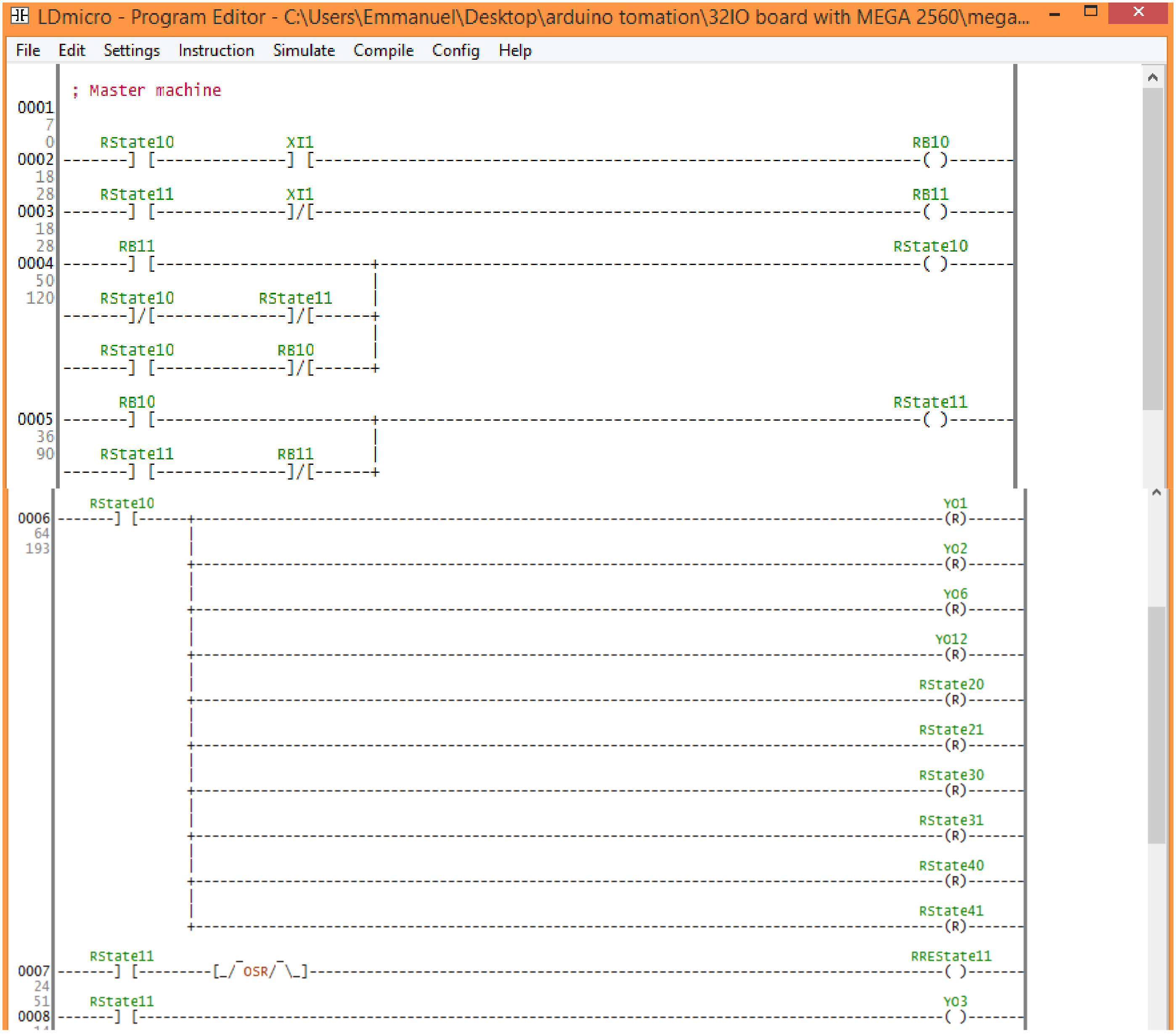
I have got some working problems with LDmicro to Mega2560 pin maping:

-PE0 and PE1 for I1 and I2 don't run as digital I/O even with some limited port manipulation on the software. So I used a switch to link this pins to PC1 and PD7

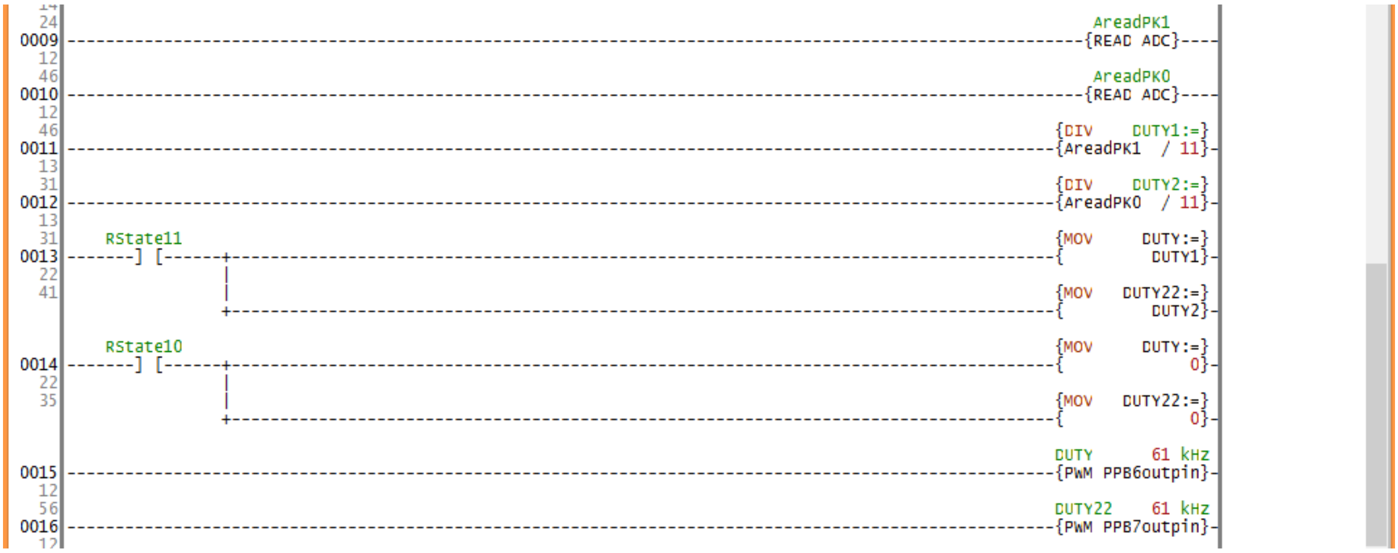
-PF0, PF1 and PF7 don't run as outputs even with some limited port manipulation on the software. So I link with a wire: PK0 to PF0, PK1 to PF1 and PK2 to PF7.

-and then, PL5 link to PB6 and PL3 link to PB7 because LDmicro doesn't allow you to use PWM on PL5 and PL3 ??!!

I choose on purpose 4 different timers with no link to demonstrate the multitasking process.


LDmicro - Program Editor - C:\Users\Emmanuel\Desktop\arduino tomation\32IO board with MEGA 2560\mega...
File
Edit
Settings
Instruction
Simulate
Compile
Config
Help

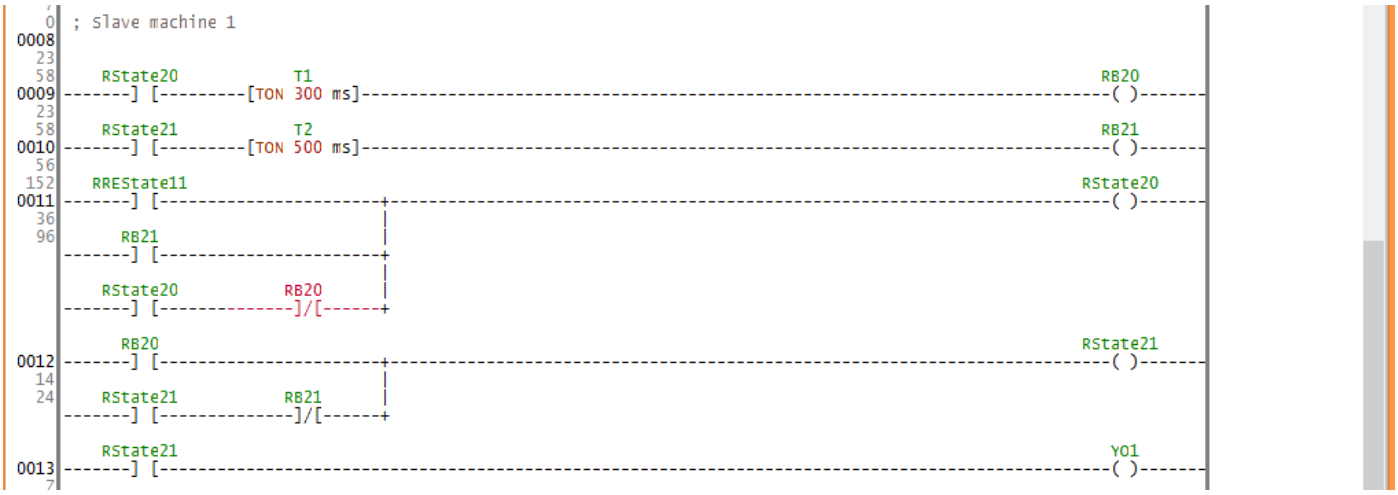
```
     ; Master machine
0001
   7
   0     RState10          XI1                                RB10
0002 -------] [------------] [--------------------------------( )-----
  18
  28     RState11          XI1                                RB11
0003 -------] [------------]/[--------------------------------( )-----
  18
  28       RB11                                               RState10
0004 -------] [----------------------+------------------------( )-----
  50                                 |
 120     RState10        RState11    |
     -------]/[------------]/[-------+
                                     |
         RState10          RB10      |
     -------] [------------]/[-------+

           RB10                                               RState11
0005 -------] [----------------------+------------------------( )-----
  36                                 |
  90     RState11          RB11      |
     -------] [------------]/[-------+

        RState10                                              YO1
0006 ------] [----+-------------------------------------------(R)-----
  64              |
 193              |                                           YO2
                  +-------------------------------------------(R)-----
                  |                                           YO6
                  +-------------------------------------------(R)-----
                  |                                           YO12
                  +-------------------------------------------(R)-----
                  |                                           RState20
                  +-------------------------------------------(R)-----
                  |                                           RState21
                  +-------------------------------------------(R)-----
                  |                                           RState30
                  +-------------------------------------------(R)-----
                  |                                           RState31
                  +-------------------------------------------(R)-----
                  |                                           RState40
                  +-------------------------------------------(R)-----
                  |                                           RState41
                  +-------------------------------------------(R)-----

        RState11                                              RREState11
0007 ------] [---------[_/¯OSR/¯\_]---------------------------( )-----
  24
  51    RState11                                              YO3
0008 ------] [------------------------------------------------( )-----
```



```
  24                                                                    AreadPK1
0009 -----------------------------------------------------------------{READ ADC}----
  12
  46                                                                    AreadPK0
0010 -----------------------------------------------------------------{READ ADC}----
  12
  46                                                                 {DIV     DUTY1:=}
0011 ----------------------------------------------------------------{AreadPK1   / 11}-
  13
  31                                                                 {DIV     DUTY2:=}
0012 ----------------------------------------------------------------{AreadPK0   / 11}-
  13
  31     RState11                                                    {MOV      DUTY:=}
0013 -------] [-------+--------------------------------------------{          DUTY1}-
  22                  |
  41                  |                                              {MOV    DUTY22:=}
                      +--------------------------------------------{          DUTY2}-

         RState10                                                    {MOV      DUTY:=}
0014 -------] [-------+--------------------------------------------{              0}-
  22                  |
  35                  |                                              {MOV    DUTY22:=}
                      +--------------------------------------------{              0}-

                                                                     DUTY       61 kHz
0015 ----------------------------------------------------------------{PWM PPB6outpin}-
  12
  56                                                                 DUTY22     61 kHz
0016 ----------------------------------------------------------------{PWM PPB7outpin}-
```

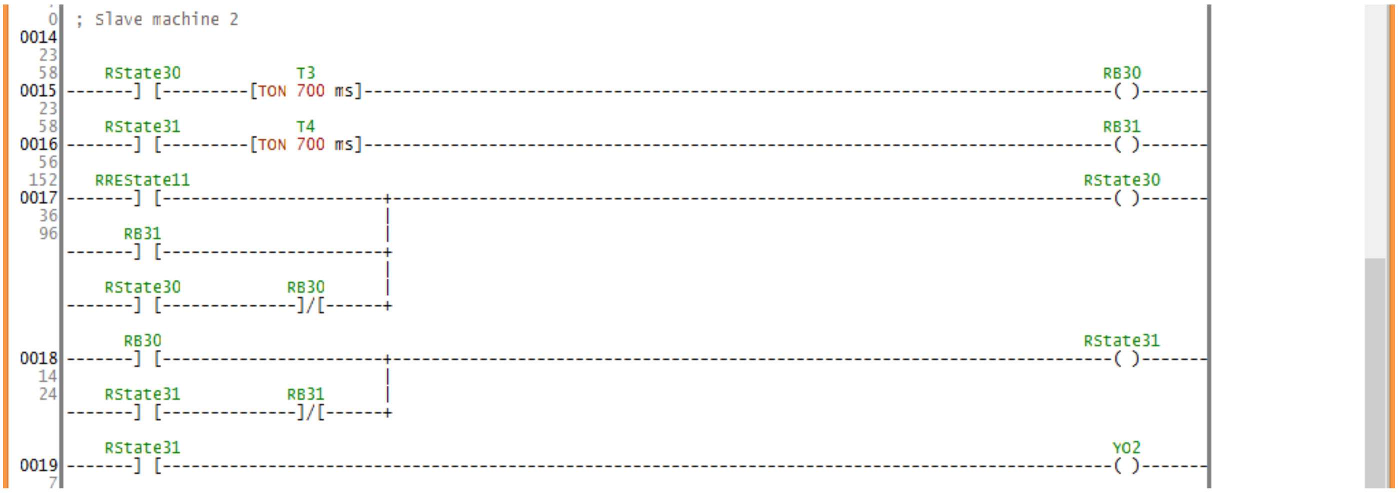
On State10 every output must be reset and all the states of the slave machines too.
A rising edge on State11 event (OSR function) launches the 3 slaves.
I use 2 analog inputs POT1 on PK0 and POT2 on PK1 to control PB6 linked to PL5 and PB7 linked to PL3.

The machine 1: blink O1



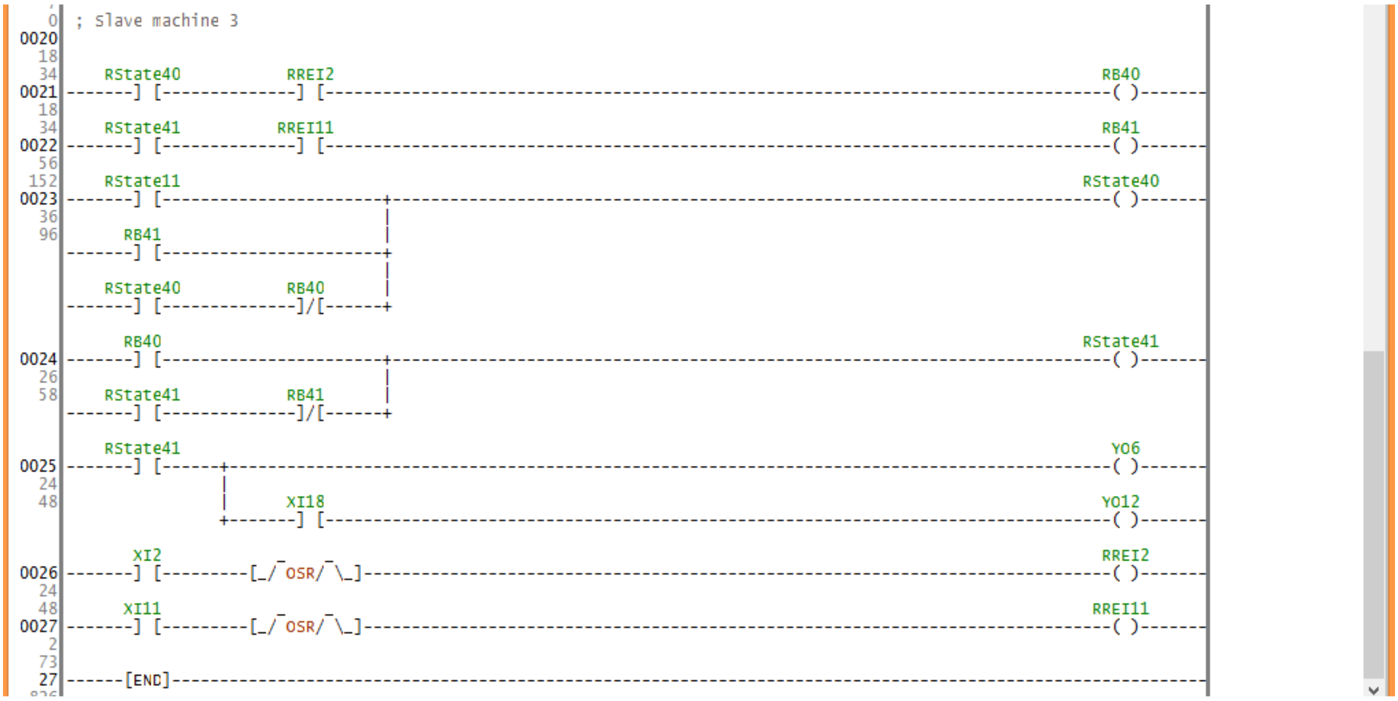
```
   0  ; Slave machine 1
0008
  23
  58     RState20           T1                                         RB20
0009 -------] [---------[TON 300 ms]---------------------------------( )-------
  23
  58     RState21           T2                                         RB21
0010 -------] [---------[TON 500 ms]---------------------------------( )-------
  56
 152   RREState11                                                     RState20
0011 -------] [-----------------------+------------------------------( )-------
  36                                  |
  96     RB21                         |
     -------] [-----------------------+
                                      |
         RState20           RB20      |
     -------] [--------------]/[------+

         RB20                                                         RState21
0012 -------] [-----------------------+------------------------------( )-------
  14                                  |
  24     RState21           RB21      |
     -------] [--------------]/[------+

         RState21                                                      YO1
0013 -------] [------------------------------------------------------( )-------
```

The machine 2: blink O2



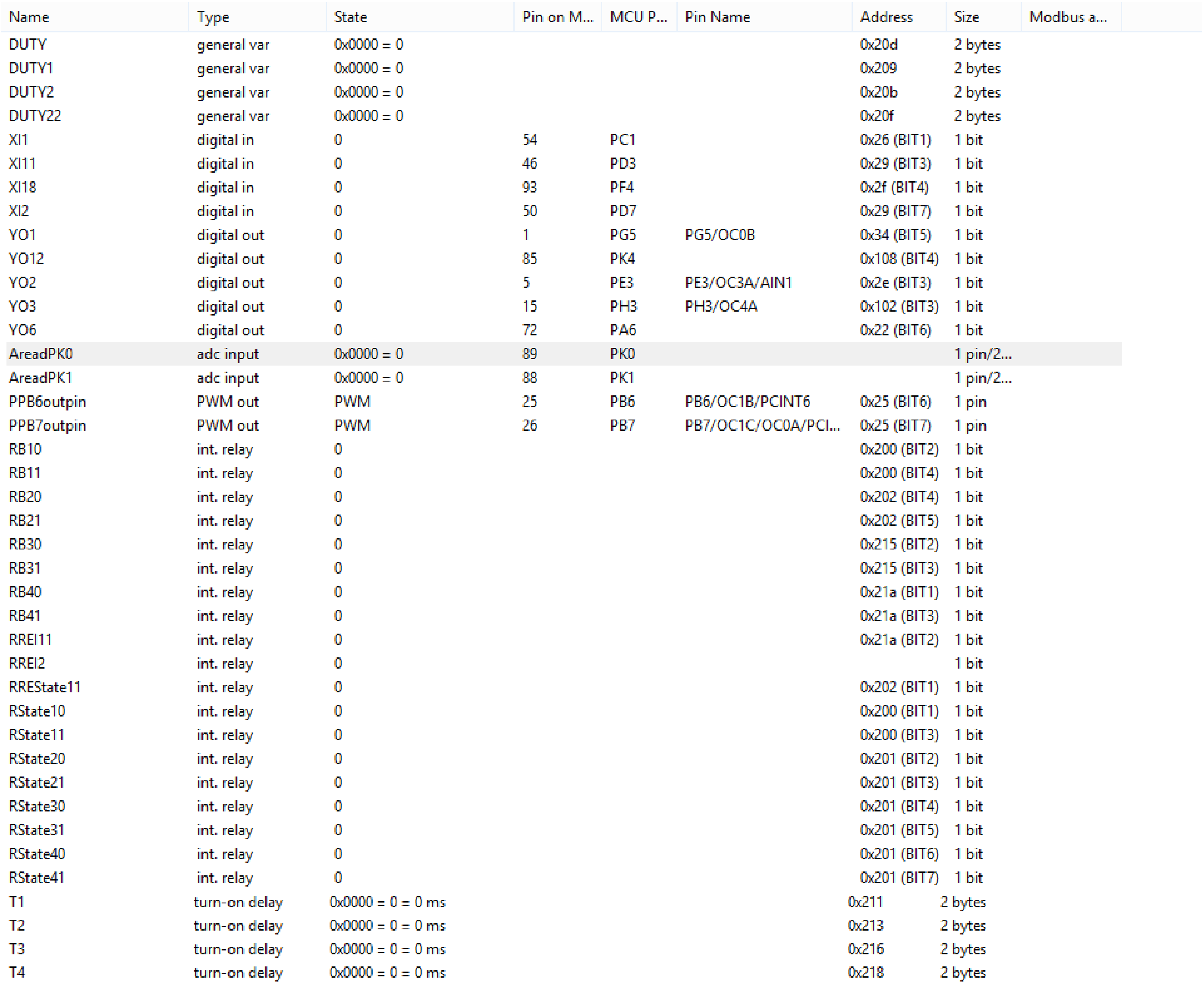
```
   0  ; Slave machine 2
0014
  23
  58     RState30           T3                                         RB30
0015 -------] [---------[TON 700 ms]---------------------------------( )-------
  23
  58     RState31           T4                                         RB31
0016 -------] [---------[TON 700 ms]---------------------------------( )-------
  56
 152   RREState11                                                     RState30
0017 -------] [-----------------------+------------------------------( )-------
  36                                  |
  96     RB31                         |
     -------] [-----------------------+
                                      |
         RState30           RB30      |
     -------] [--------------]/[------+

         RB30                                                         RState31
0018 -------] [-----------------------+------------------------------( )-------
  14                                  |
  24     RState31           RB31      |
     -------] [--------------]/[------+

         RState31                                                      YO2
0019 -------] [------------------------------------------------------( )-------
```

The Machine 3: RUN/STOP O6 and switch ON O12 if I18 is pushed within the State41.



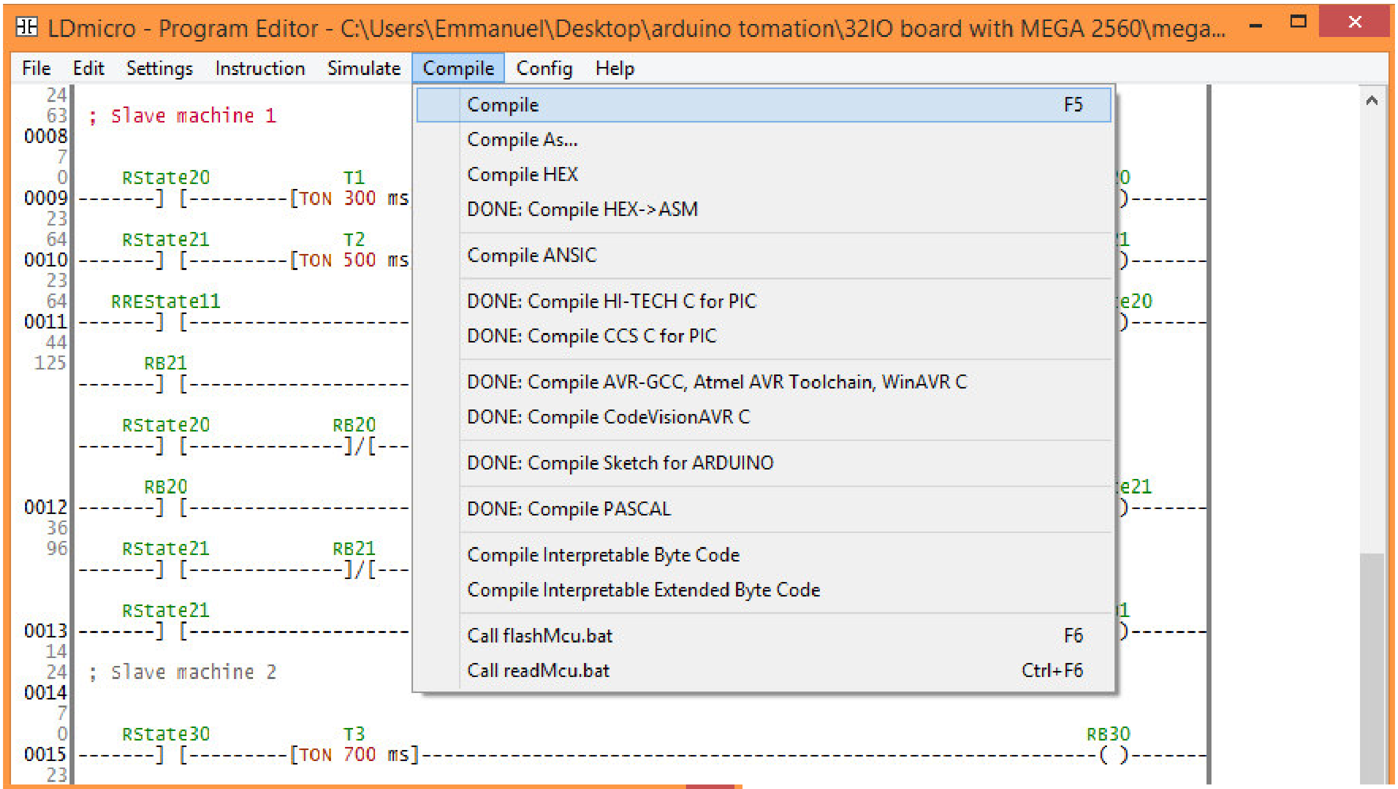
```
   0  ; Slave machine 3
0020
  18
  34    RState40          RREI2                                                    RB40
0021 -------] [--------------] [----------------------------------------------------( )-------
  18
  34    RState41         RREI11                                                    RB41
0022 -------] [--------------] [----------------------------------------------------( )-------
  56
 152    RState11                                                               RState40
0023 -------] [----------------------+----------------------------------------------( )-------
  36                                 |
  96      RB41                       |
     -------] [----------------------+
                                     |
        RState40          RB40       |
     -------] [--------------]/[-----+

          RB40                                                                 RState41
0024 -------] [----------------------+----------------------------------------------( )-------
  26                                 |
  58    RState41          RB41       |
     -------] [--------------]/[-----+

        RState41                                                                    Y06
0025 -------] [------+------------------------------------------------------------( )-------
  24                 |
  48                 |    XI18                                                     Y012
                     +-------] [--------------------------------------------------( )-------

           XI2                                                                     RREI2
0026 -------] [---------[_/¯OSR/¯\_]-----------------------------------------------( )-------
  24
  48      XI11                                                                    RREI11
0027 -------] [---------[_/¯OSR/¯\_]-----------------------------------------------( )-------
   2
  73
  27 ------[END]-----------------------------------------------------------------------------
```

The list of INPUT/OUTPUT addresses on the microcontroller and the internal relays used in the ladder.

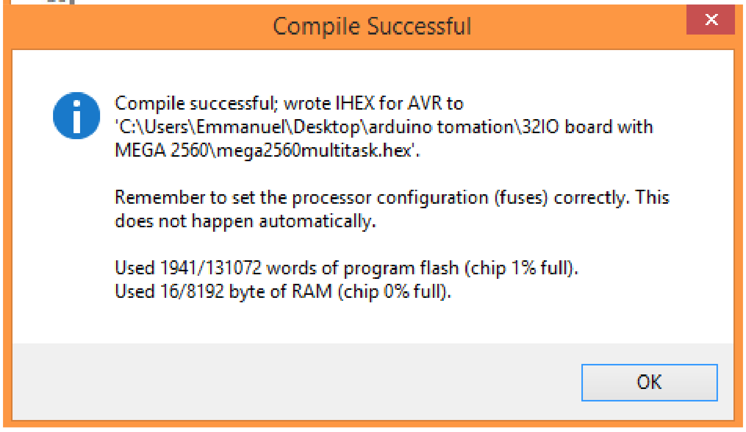


Name	Type	State	Pin on M...	MCU P...	Pin Name	Address	Size	Modbus a...
DUTY	general var	0x0000 = 0				0x20d	2 bytes	
DUTY1	general var	0x0000 = 0				0x209	2 bytes	
DUTY2	general var	0x0000 = 0				0x20b	2 bytes	
DUTY22	general var	0x0000 = 0				0x20f	2 bytes	
XI1	digital in	0	54	PC1		0x26 (BIT1)	1 bit	
XI11	digital in	0	46	PD3		0x29 (BIT3)	1 bit	
XI18	digital in	0	93	PF4		0x2f (BIT4)	1 bit	
XI2	digital in	0	50	PD7		0x29 (BIT7)	1 bit	
YO1	digital out	0	1	PG5	PG5/OC0B	0x34 (BIT5)	1 bit	
YO12	digital out	0	85	PK4		0x108 (BIT4)	1 bit	
YO2	digital out	0	5	PE3	PE3/OC3A/AIN1	0x2e (BIT3)	1 bit	
YO3	digital out	0	15	PH3	PH3/OC4A	0x102 (BIT3)	1 bit	
YO6	digital out	0	72	PA6		0x22 (BIT6)	1 bit	
AreadPK0	adc input	0x0000 = 0	89	PK0			1 pin/2...	
AreadPK1	adc input	0x0000 = 0	88	PK1			1 pin/2...	
PPB6outpin	PWM out	PWM	25	PB6	PB6/OC1B/PCINT6	0x25 (BIT6)	1 pin	
PPB7outpin	PWM out	PWM	26	PB7	PB7/OC1C/OC0A/PCI...	0x25 (BIT7)	1 pin	
RB10	int. relay	0				0x200 (BIT2)	1 bit	
RB11	int. relay	0				0x200 (BIT4)	1 bit	
RB20	int. relay	0				0x202 (BIT4)	1 bit	
RB21	int. relay	0				0x202 (BIT5)	1 bit	
RB30	int. relay	0				0x215 (BIT2)	1 bit	
RB31	int. relay	0				0x215 (BIT3)	1 bit	
RB40	int. relay	0				0x21a (BIT1)	1 bit	
RB41	int. relay	0				0x21a (BIT3)	1 bit	
RREI11	int. relay	0				0x21a (BIT2)	1 bit	
RREI2	int. relay	0					1 bit	
RREState11	int. relay	0				0x202 (BIT1)	1 bit	
RState10	int. relay	0				0x200 (BIT1)	1 bit	
RState11	int. relay	0				0x200 (BIT3)	1 bit	
RState20	int. relay	0				0x201 (BIT2)	1 bit	
RState21	int. relay	0				0x201 (BIT3)	1 bit	
RState30	int. relay	0				0x201 (BIT4)	1 bit	
RState31	int. relay	0				0x201 (BIT5)	1 bit	
RState40	int. relay	0				0x201 (BIT6)	1 bit	
RState41	int. relay	0				0x201 (BIT7)	1 bit	
T1	turn-on delay	0x0000 = 0 = 0 ms				0x211	2 bytes	
T2	turn-on delay	0x0000 = 0 = 0 ms				0x213	2 bytes	
T3	turn-on delay	0x0000 = 0 = 0 ms				0x216	2 bytes	
T4	turn-on delay	0x0000 = 0 = 0 ms				0x218	2 bytes	

1-3 Compile the program in .hex file:


LDmicro - Program Editor - C:\Users\Emmanuel\Desktop\arduino tomation\32IO board with MEGA 2560\mega...
File
Edit
Settings
Instruction
Simulate
Compile
Config
Help
Compile F5
Compile As...
Compile HEX
DONE: Compile HEX->ASM
Compile ANSIC
DONE: Compile HI-TECH C for PIC
DONE: Compile CCS C for PIC
DONE: Compile AVR-GCC, Atmel AVR Toolchain, WinAVR C
DONE: Compile CodeVisionAVR C
DONE: Compile Sketch for ARDUINO
DONE: Compile PASCAL
Compile Interpretable Byte Code
Compile Interpretable Extended Byte Code
Call flashMcu.bat F6
Call readMcu.bat Ctrl+F6

```
  24
  63  ; Slave machine 1
0008
   7
   0    RState20          T1
0009 -------] [---------[TON 300 ms
  23
  64    RState21          T2
0010 -------] [---------[TON 500 ms
  23
  64   RREState11
0011 -------] [--------------------
  44
 125      RB21
     -------] [--------------------

        RState20         RB20
     -------] [--------------]/[---

          RB20
0012 -------] [--------------------
  36
  96    RState21         RB21
     -------] [--------------]/[---

        RState21
0013 -------] [--------------------
  14
  24  ; Slave machine 2
0014
   7
   0    RState30          T3                                                          RB30
0015 -------] [---------[TON 700 ms]------------------------------------------------( )-------
  23
```



Compile successful; wrote IHEX for AVR to 'C:\Users\Emmanuel\Desktop\arduino tomation\32IO board with MEGA 2560\mega2560multitask.hex'.

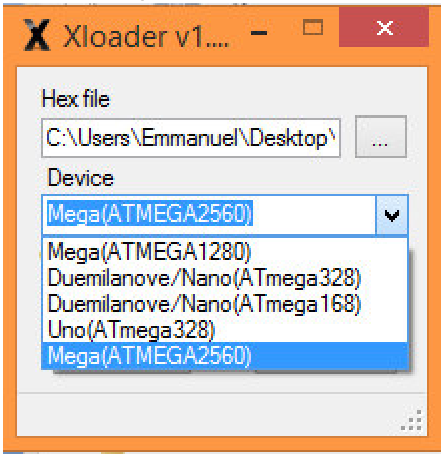
Remember to set the processor configuration (fuses) correctly. This does not happen automatically.

Used 1941/131072 words of program flash (chip 1% full).
Used 16/8192 byte of RAM (chip 0% full).

1-4 Launch Xloader downloader:

Because the Mega board is uploaded with an USB wire not USBasp, you can't use Kazhama anymore to download the .hex file. Xloader does the job.

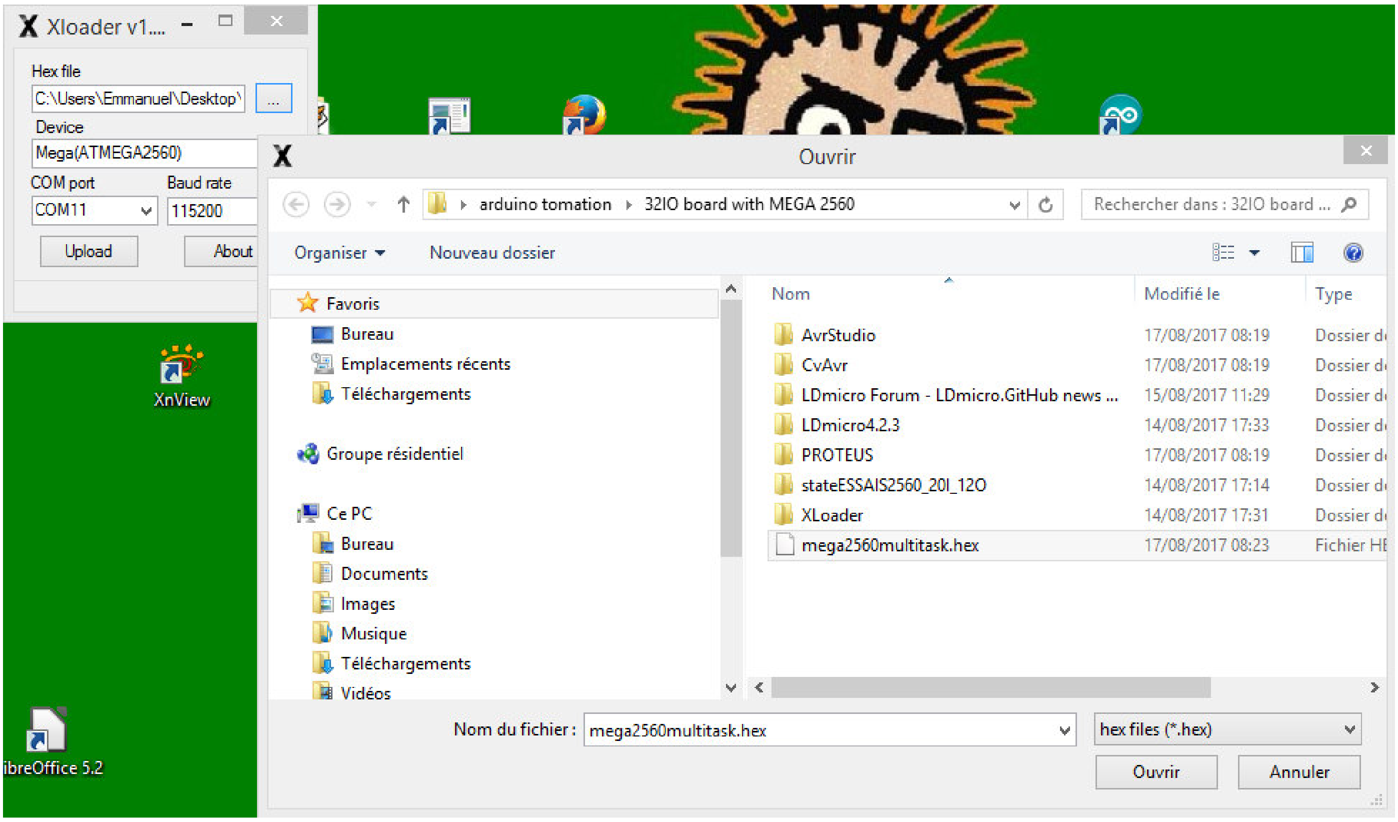
Select the good Microcontroller:



Hex file: C:\Users\Emmanuel\Desktop\

Device: Mega(ATMEGA2560)

Load the hex file you have just created:



Hex file: C:\Users\Emmanuel\Desktop\
Device: Mega(ATMEGA2560)
COM port: COM11
Baud rate: 115200

arduino tomation › 32IO board with MEGA 2560

Rechercher dans : 32IO board ...

Favoris
- Bureau
- Emplacements récents
- Téléchargements

Groupe résidentiel

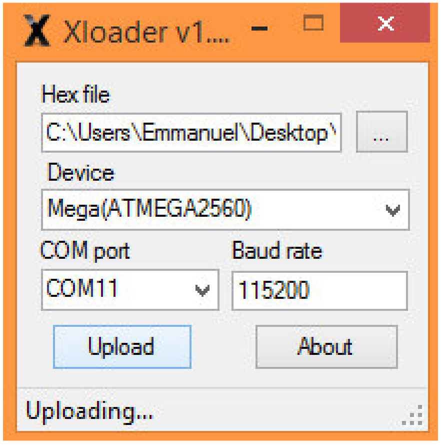
Ce PC
- Bureau
- Documents
- Images
- Musique
- Téléchargements
- Vidéos

Nom	Modifié le	Type
AvrStudio	17/08/2017 08:19	Dossier d
CvAvr	17/08/2017 08:19	Dossier d
LDmicro Forum - LDmicro.GitHub news ...	15/08/2017 11:29	Dossier d
LDmicro4.2.3	14/08/2017 17:33	Dossier d
PROTEUS	17/08/2017 08:19	Dossier d
stateESSAIS2560_20I_12O	14/08/2017 17:14	Dossier d
XLoader	14/08/2017 17:31	Dossier d
mega2560multitask.hex	17/08/2017 08:23	Fichier HE

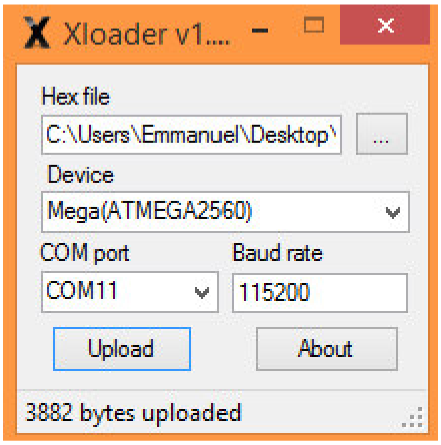
Nom du fichier : mega2560multitask.hex
hex files (*.hex)

Switch off the 32I/O board on RUN (to connect the pins D10, D11, D12 and D13 to the digital outputs, here they are not the SPI bus).

Select the good COM port and Upload:



Hex file: C:\Users\Emmanuel\Desktop\
Device: Mega(ATMEGA2560)
COM port: COM11
Baud rate: 115200
Uploading...



Hex file: C:\Users\Emmanuel\Desktop\
Device: Mega(ATMEGA2560)
COM port: COM11
Baud rate: 115200
3882 bytes uploaded

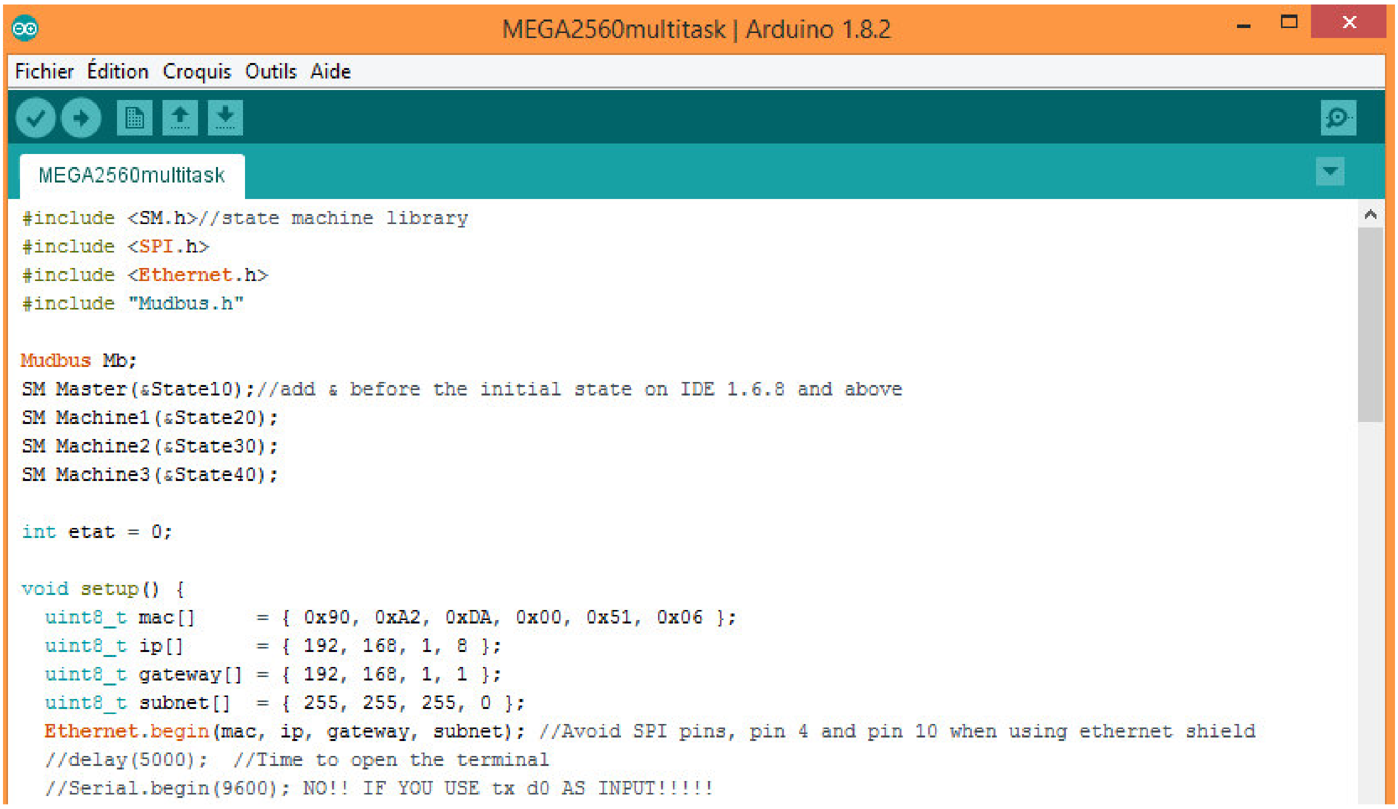
That’s it and enjoy.

2-Multitasking with programming on Arduino IDE 1.8.2 using SM library:

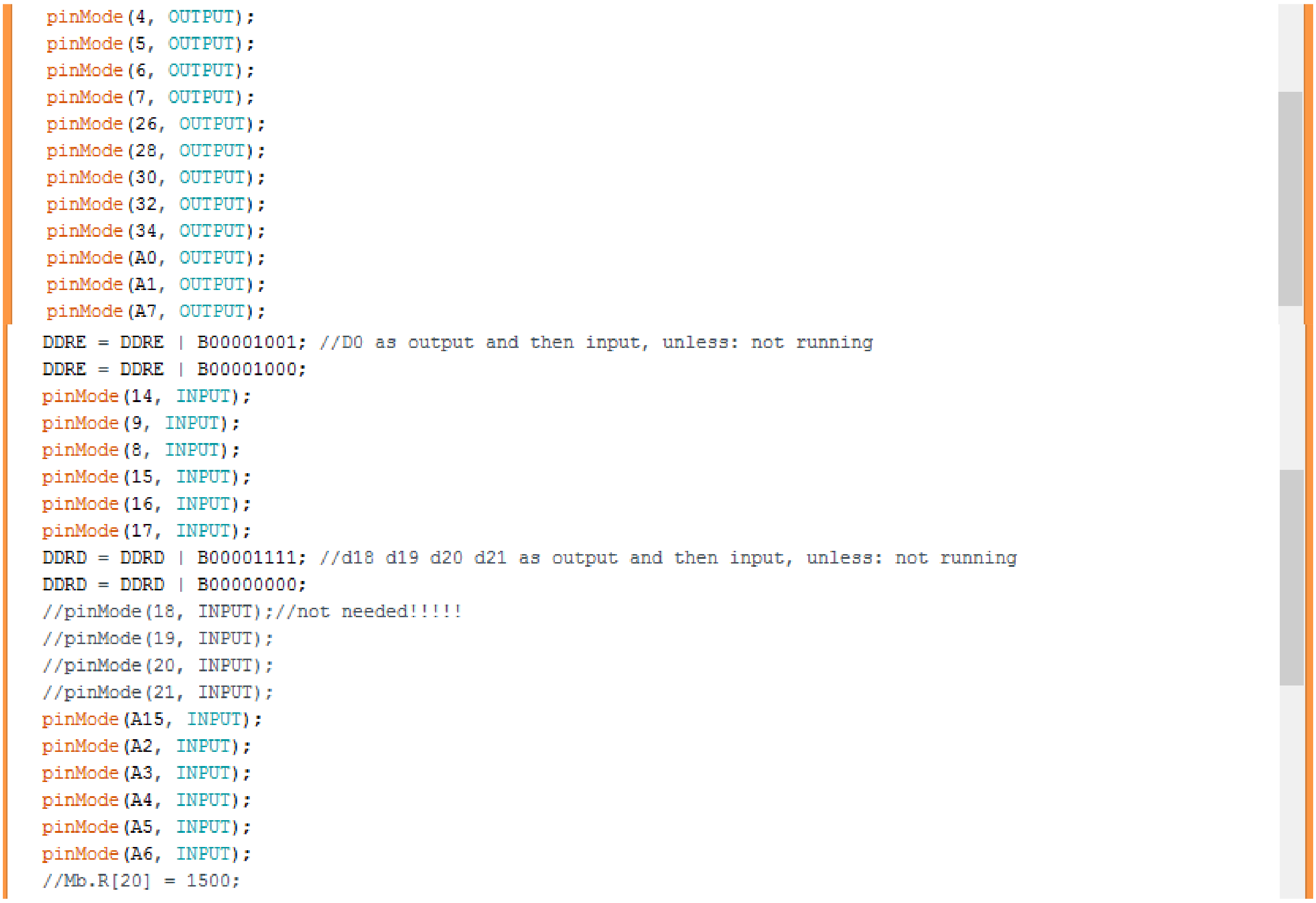
In order to use pins 0 and pin 1 on the MEGA 2560 board you need to manipulate the port E with this trick:

```
DDRE = DDRE | B00001001; //D0 as output and then input, unless: not running
DDRE = DDRE | B00001000;
```

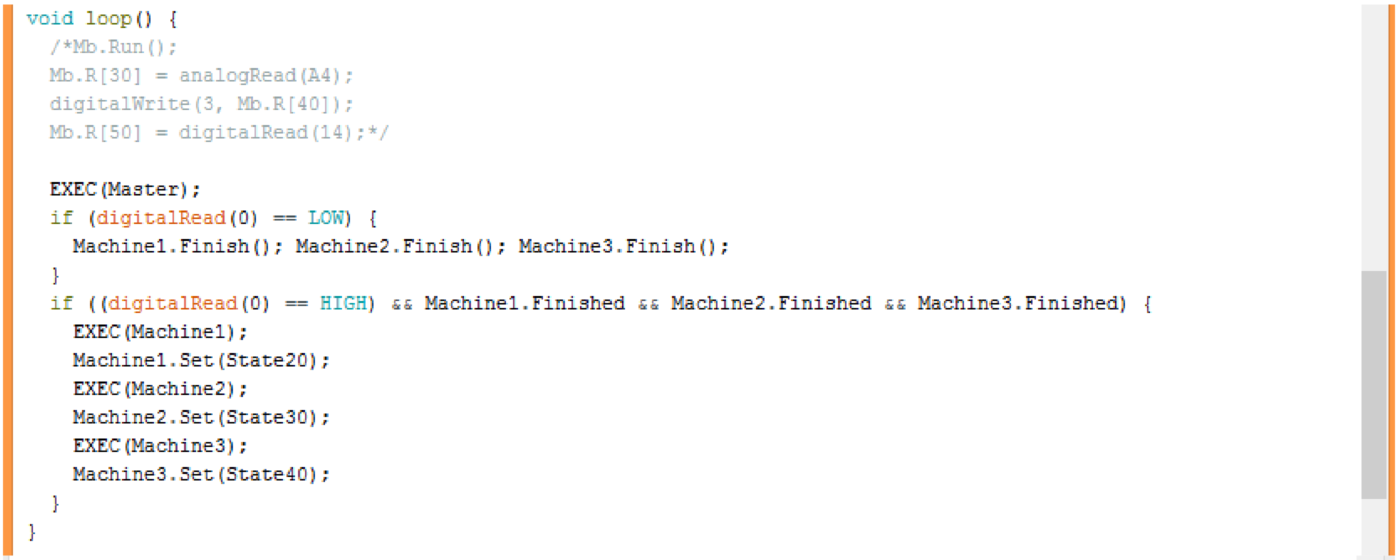
You need also to disable Serial communication, disconnect all the links needed on LDmicro and put the switches as advised in the supplied guide.


MEGA2560multitask | Arduino 1.8.2
Fichier Édition Croquis Outils Aide
MEGA2560multitask

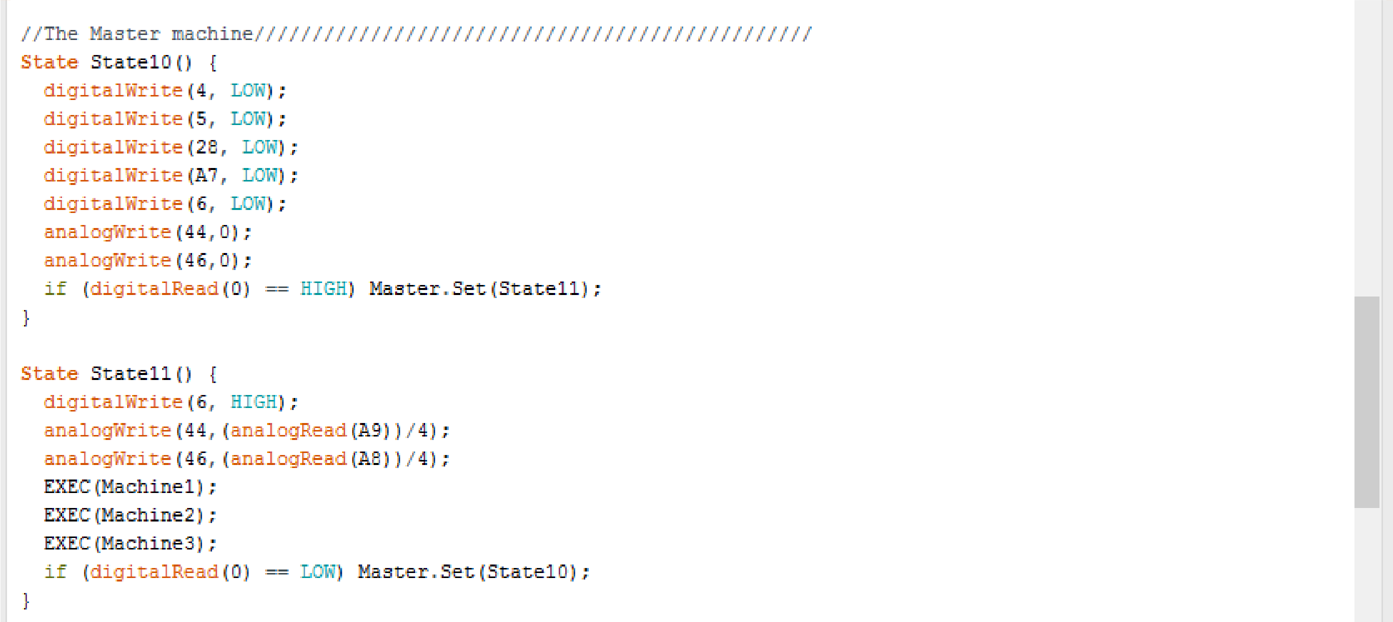
```
#include <SM.h>//state machine library
#include <SPI.h>
#include <Ethernet.h>
#include "Mudbus.h"

Mudbus Mb;
SM Master(&State10);//add & before the initial state on IDE 1.6.8 and above
SM Machine1(&State20);
SM Machine2(&State30);
SM Machine3(&State40);

int etat = 0;

void setup() {
  uint8_t mac[]     = { 0x90, 0xA2, 0xDA, 0x00, 0x51, 0x06 };
  uint8_t ip[]      = { 192, 168, 1, 8 };
  uint8_t gateway[] = { 192, 168, 1, 1 };
  uint8_t subnet[]  = { 255, 255, 255, 0 };
  Ethernet.begin(mac, ip, gateway, subnet); //Avoid SPI pins, pin 4 and pin 10 when using ethernet shield
  //delay(5000);  //Time to open the terminal
  //Serial.begin(9600); NO!! IF YOU USE tx d0 AS INPUT!!!!!
```



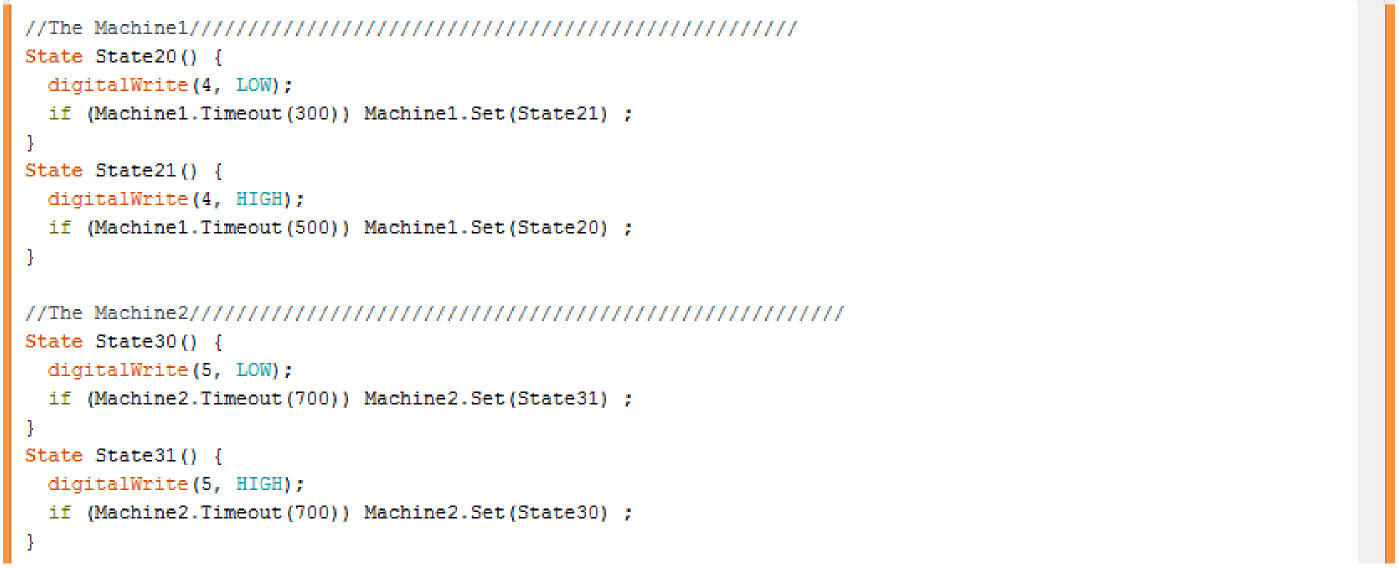
```
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(26, OUTPUT);
  pinMode(28, OUTPUT);
  pinMode(30, OUTPUT);
  pinMode(32, OUTPUT);
  pinMode(34, OUTPUT);
  pinMode(A0, OUTPUT);
  pinMode(A1, OUTPUT);
  pinMode(A7, OUTPUT);
  DDRE = DDRE | B00001001; //D0 as output and then input, unless: not running
  DDRE = DDRE | B00001000;
  pinMode(14, INPUT);
  pinMode(9, INPUT);
  pinMode(8, INPUT);
  pinMode(15, INPUT);
  pinMode(16, INPUT);
  pinMode(17, INPUT);
  DDRD = DDRD | B00001111; //d18 d19 d20 d21 as output and then input, unless: not running
  DDRD = DDRD | B00000000;
  //pinMode(18, INPUT);//not needed!!!!!
  //pinMode(19, INPUT);
  //pinMode(20, INPUT);
  //pinMode(21, INPUT);
  pinMode(A15, INPUT);
  pinMode(A2, INPUT);
  pinMode(A3, INPUT);
  pinMode(A4, INPUT);
  pinMode(A5, INPUT);
  pinMode(A6, INPUT);
  //Mb.R[20] = 1500;
```



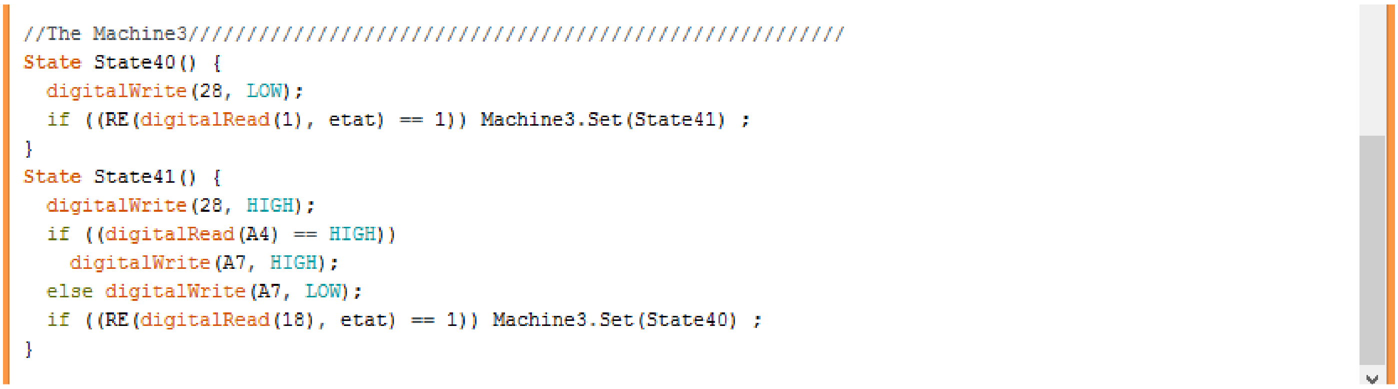
```
void loop() {
  /*Mb.Run();
  Mb.R[30] = analogRead(A4);
  digitalWrite(3, Mb.R[40]);
  Mb.R[50] = digitalRead(14);*/

  EXEC(Master);
  if (digitalRead(0) == LOW) {
    Machine1.Finish(); Machine2.Finish(); Machine3.Finish();
  }
  if ((digitalRead(0) == HIGH) && Machine1.Finished && Machine2.Finished && Machine3.Finished) {
    EXEC(Machine1);
    Machine1.Set(State20);
    EXEC(Machine2);
    Machine2.Set(State30);
    EXEC(Machine3);
    Machine3.Set(State40);
  }
}
```



```
//The Master machine///////////////////////////////////////////////
State State10() {
  digitalWrite(4, LOW);
  digitalWrite(5, LOW);
  digitalWrite(28, LOW);
  digitalWrite(A7, LOW);
  digitalWrite(6, LOW);
  analogWrite(44,0);
  analogWrite(46,0);
  if (digitalRead(0) == HIGH) Master.Set(State11);
}

State State11() {
  digitalWrite(6, HIGH);
  analogWrite(44,(analogRead(A9))/4);
  analogWrite(46,(analogRead(A8))/4);
  EXEC(Machine1);
  EXEC(Machine2);
  EXEC(Machine3);
  if (digitalRead(0) == LOW) Master.Set(State10);
}
```



```
//The Machine1/////////////////////////////////////////////////////
State State20() {
  digitalWrite(4, LOW);
  if (Machine1.Timeout(300)) Machine1.Set(State21) ;
}
State State21() {
  digitalWrite(4, HIGH);
  if (Machine1.Timeout(500)) Machine1.Set(State20) ;
}

//The Machine2/////////////////////////////////////////////////////////
State State30() {
  digitalWrite(5, LOW);
  if (Machine2.Timeout(700)) Machine2.Set(State31) ;
}
State State31() {
  digitalWrite(5, HIGH);
  if (Machine2.Timeout(700)) Machine2.Set(State30) ;
}
```



```
//The Machine3//////////////////////////////////////////////////////////
State State40() {
  digitalWrite(28, LOW);
  if ((RE(digitalRead(1), etat) == 1)) Machine3.Set(State41) ;
}
State State41() {
  digitalWrite(28, HIGH);
  if ((digitalRead(A4) == HIGH))
    digitalWrite(A7, HIGH);
  else digitalWrite(A7, LOW);
  if ((RE(digitalRead(18), etat) == 1)) Machine3.Set(State40) ;
}
```

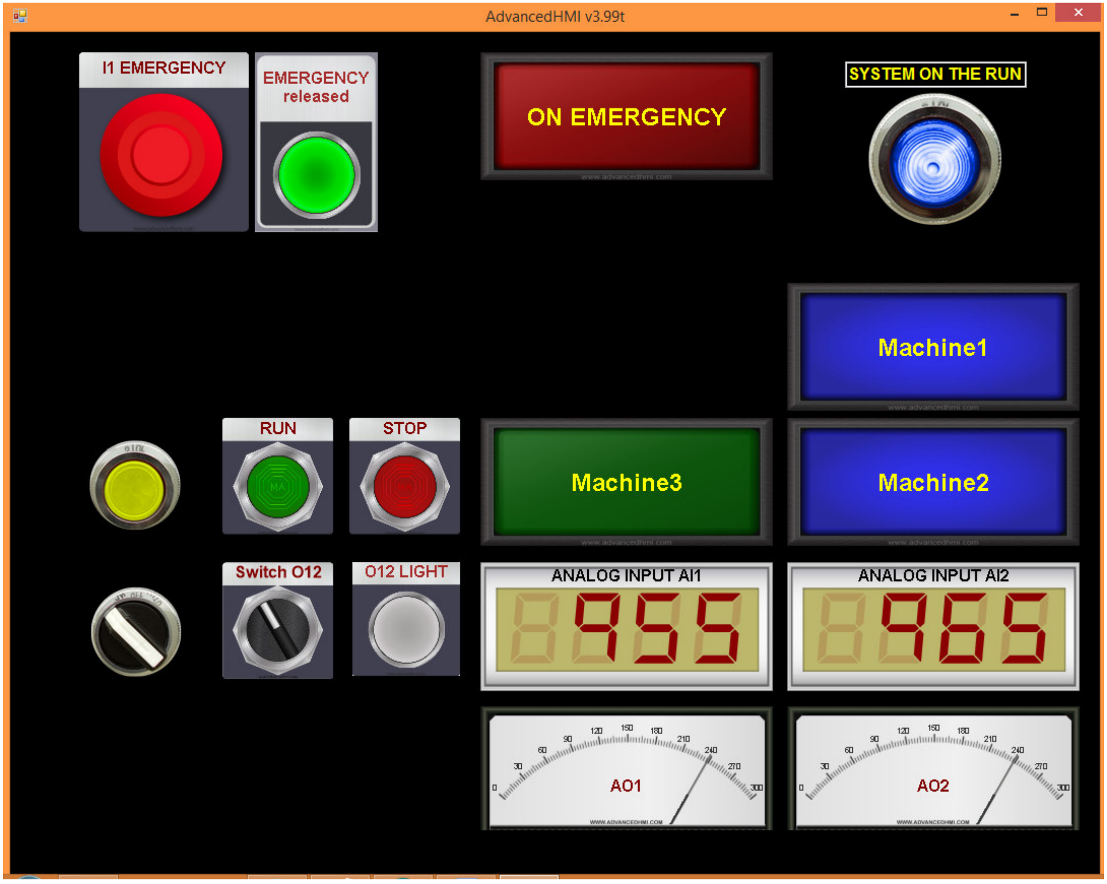
3-Multitasking with programming on Arduino IDE 1.8.2 using SM library and supervising on AdvancedHMI:

You need to switch on O1 to pin24 in order to disconnect D4, a pin used by the Ethernet shield. The shield is only connected to the MEGA 2560 board by:

-pin D4
-pin D10
-ICSP connector (SPI bus, GND, +5V).

The control panel is made of:

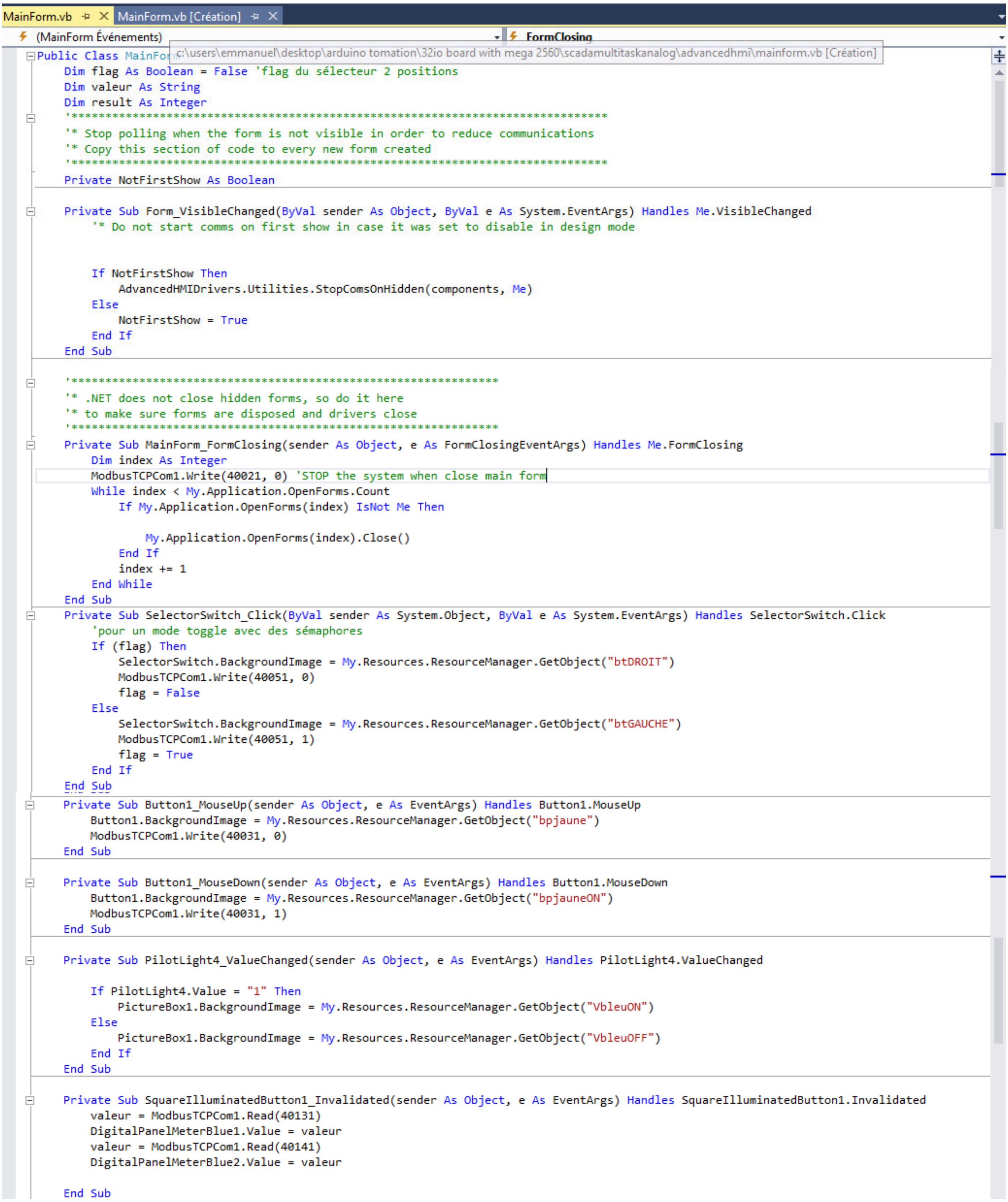
-an Emergency mushroom head button
-a light to know the emergency button state
-a light to know if we are on emergency state
-a blinking light for Machine1
-a blinking light for Machine2
-a light for Machine3 switched ON/OFF with the push-buttons RUN and STOP
-a light switched ON/OFF with a selector switch during the run of Machine3 only
-2 digital panels meter for 2 analog inputs
-2 gauges for 2 analog outputs



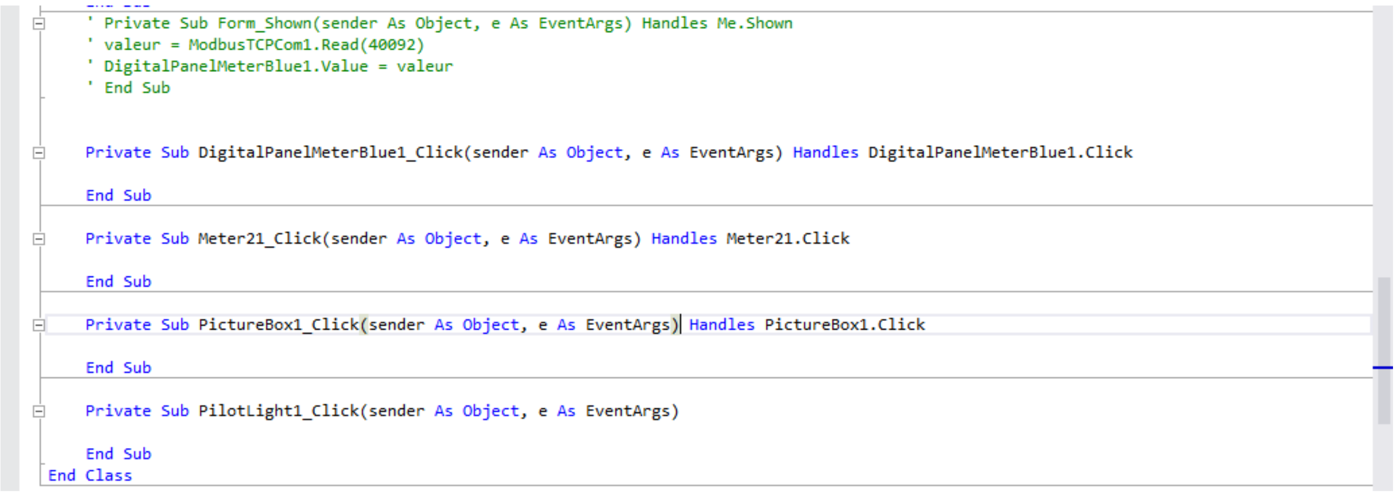
When you push the emergency: the master machine stays in state 10 (Reset of all the system) but it's not safe because at the same time the power must be switch off on the actuator (EMERGENCY RULE) and it's only done with the real mushroom push-button.

When you close the mainform of advancedHMI, the system is reset on state10.

The modified code of the Mainform:

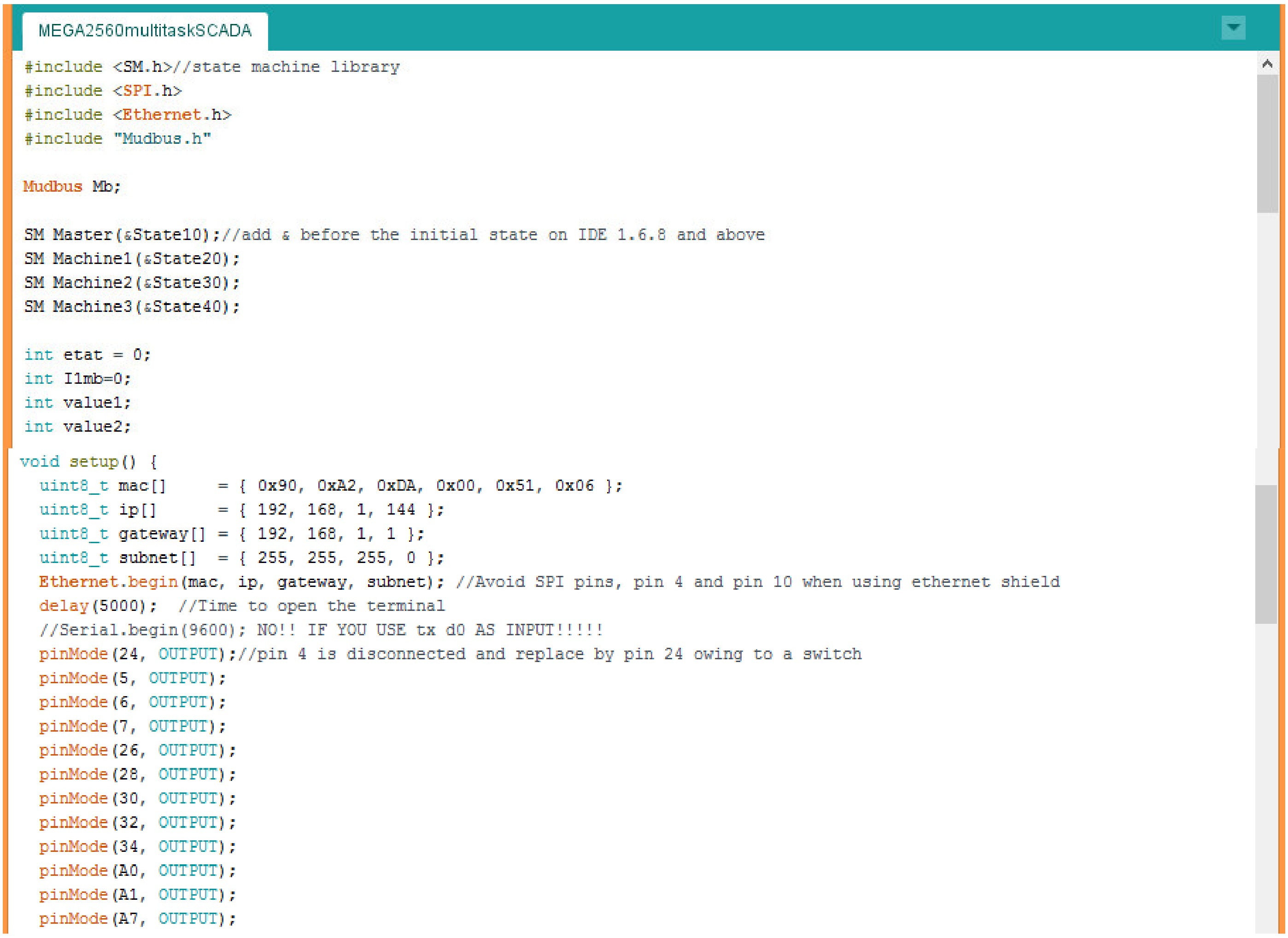

MainForm.vb
MainForm.vb [Création]
(MainForm Événements)
FormClosing
c:\users\emmanuel\desktop\arduino tomation\32io board with mega 2560\scadamultitaskanalog\advancedhmi\mainform.vb [Création]

```vb
Public Class MainForm
    Dim flag As Boolean = False 'flag du sélecteur 2 positions
    Dim valeur As String
    Dim result As Integer
    '*******************************************************************************
    '* Stop polling when the form is not visible in order to reduce communications
    '* Copy this section of code to every new form created
    '*******************************************************************************
    Private NotFirstShow As Boolean

    Private Sub Form_VisibleChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.VisibleChanged
        '* Do not start comms on first show in case it was set to disable in design mode


        If NotFirstShow Then
            AdvancedHMIDrivers.Utilities.StopComsOnHidden(components, Me)
        Else
            NotFirstShow = True
        End If
    End Sub

    '***************************************************************
    '* .NET does not close hidden forms, so do it here
    '* to make sure forms are disposed and drivers close
    '***************************************************************
    Private Sub MainForm_FormClosing(sender As Object, e As FormClosingEventArgs) Handles Me.FormClosing
        Dim index As Integer
        ModbusTCPCom1.Write(40021, 0) 'STOP the system when close main form
        While index < My.Application.OpenForms.Count
            If My.Application.OpenForms(index) IsNot Me Then

                My.Application.OpenForms(index).Close()
            End If
            index += 1
        End While
    End Sub
    Private Sub SelectorSwitch_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SelectorSwitch.Click
        'pour un mode toggle avec des sémaphores
        If (flag) Then
            SelectorSwitch.BackgroundImage = My.Resources.ResourceManager.GetObject("btDROIT")
            ModbusTCPCom1.Write(40051, 0)
            flag = False
        Else
            SelectorSwitch.BackgroundImage = My.Resources.ResourceManager.GetObject("btGAUCHE")
            ModbusTCPCom1.Write(40051, 1)
            flag = True
        End If
    End Sub
    Private Sub Button1_MouseUp(sender As Object, e As EventArgs) Handles Button1.MouseUp
        Button1.BackgroundImage = My.Resources.ResourceManager.GetObject("bpjaune")
        ModbusTCPCom1.Write(40031, 0)
    End Sub

    Private Sub Button1_MouseDown(sender As Object, e As EventArgs) Handles Button1.MouseDown
        Button1.BackgroundImage = My.Resources.ResourceManager.GetObject("bpjauneON")
        ModbusTCPCom1.Write(40031, 1)
    End Sub

    Private Sub PilotLight4_ValueChanged(sender As Object, e As EventArgs) Handles PilotLight4.ValueChanged

        If PilotLight4.Value = "1" Then
            PictureBox1.BackgroundImage = My.Resources.ResourceManager.GetObject("VbleuON")
        Else
            PictureBox1.BackgroundImage = My.Resources.ResourceManager.GetObject("VbleuOFF")
        End If
    End Sub

    Private Sub SquareIlluminatedButton1_Invalidated(sender As Object, e As EventArgs) Handles SquareIlluminatedButton1.Invalidated
        valeur = ModbusTCPCom1.Read(40131)
        DigitalPanelMeterBlue1.Value = valeur
        valeur = ModbusTCPCom1.Read(40141)
        DigitalPanelMeterBlue2.Value = valeur

    End Sub
```

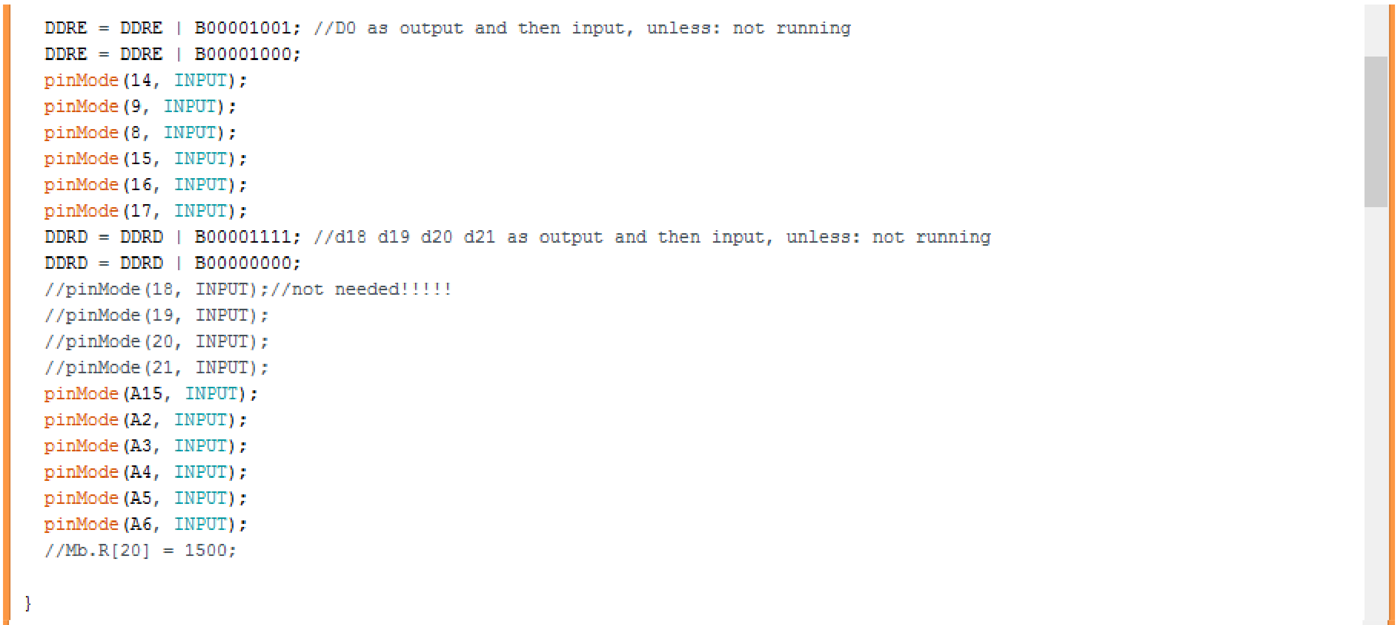


```
    ' Private Sub Form_Shown(sender As Object, e As EventArgs) Handles Me.Shown
    ' valeur = ModbusTCPCom1.Read(40092)
    ' DigitalPanelMeterBlue1.Value = valeur
    ' End Sub


    Private Sub DigitalPanelMeterBlue1_Click(sender As Object, e As EventArgs) Handles DigitalPanelMeterBlue1.Click

    End Sub

    Private Sub Meter21_Click(sender As Object, e As EventArgs) Handles Meter21.Click

    End Sub

    Private Sub PictureBox1_Click(sender As Object, e As EventArgs) Handles PictureBox1.Click

    End Sub

    Private Sub PilotLight1_Click(sender As Object, e As EventArgs)

    End Sub
End Class
```

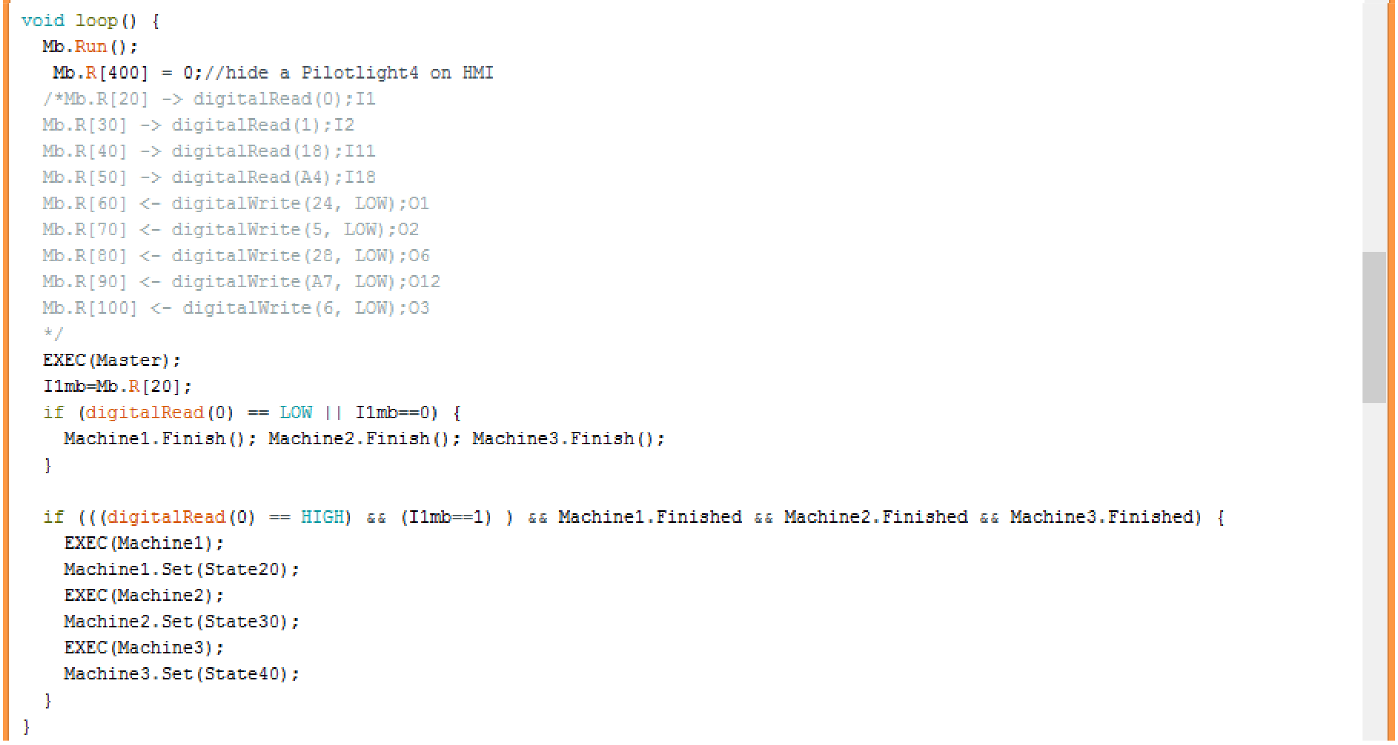
The modified arduino sketch:



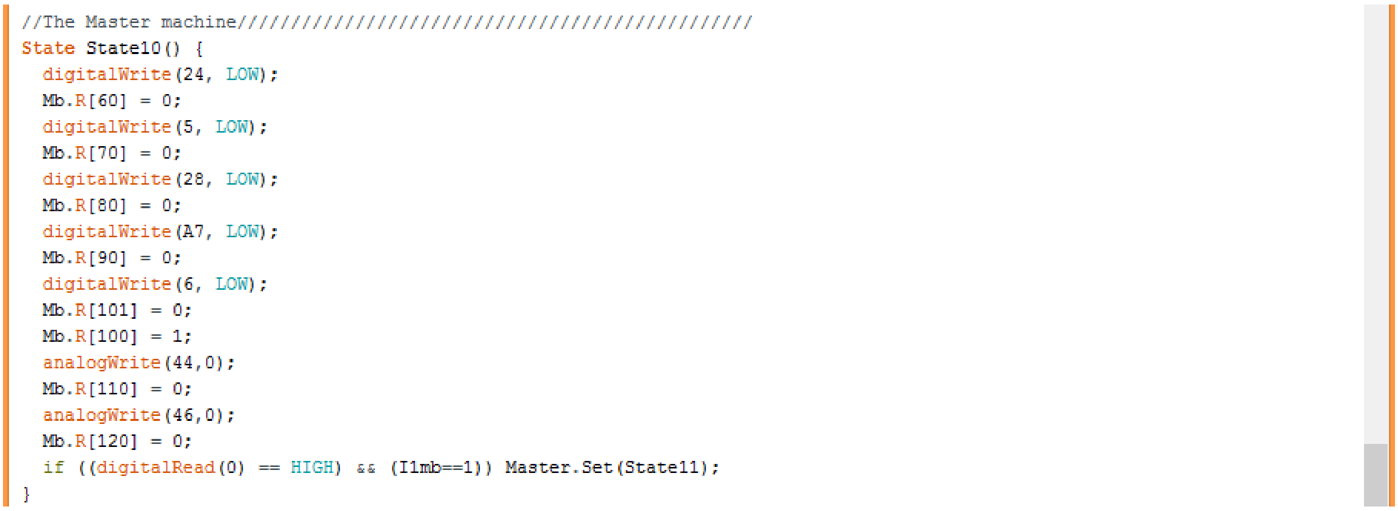
```
#include <SM.h>//state machine library
#include <SPI.h>
#include <Ethernet.h>
#include "Mudbus.h"

Mudbus Mb;

SM Master(&State10);//add & before the initial state on IDE 1.6.8 and above
SM Machine1(&State20);
SM Machine2(&State30);
SM Machine3(&State40);

int etat = 0;
int I1mb=0;
int value1;
int value2;

void setup() {
  uint8_t mac[]     = { 0x90, 0xA2, 0xDA, 0x00, 0x51, 0x06 };
  uint8_t ip[]      = { 192, 168, 1, 144 };
  uint8_t gateway[] = { 192, 168, 1, 1 };
  uint8_t subnet[]  = { 255, 255, 255, 0 };
  Ethernet.begin(mac, ip, gateway, subnet); //Avoid SPI pins, pin 4 and pin 10 when using ethernet shield
  delay(5000);  //Time to open the terminal
  //Serial.begin(9600); NO!! IF YOU USE tx d0 AS INPUT!!!!!
  pinMode(24, OUTPUT);//pin 4 is disconnected and replace by pin 24 owing to a switch
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(26, OUTPUT);
  pinMode(28, OUTPUT);
  pinMode(30, OUTPUT);
  pinMode(32, OUTPUT);
  pinMode(34, OUTPUT);
  pinMode(A0, OUTPUT);
  pinMode(A1, OUTPUT);
  pinMode(A7, OUTPUT);
```



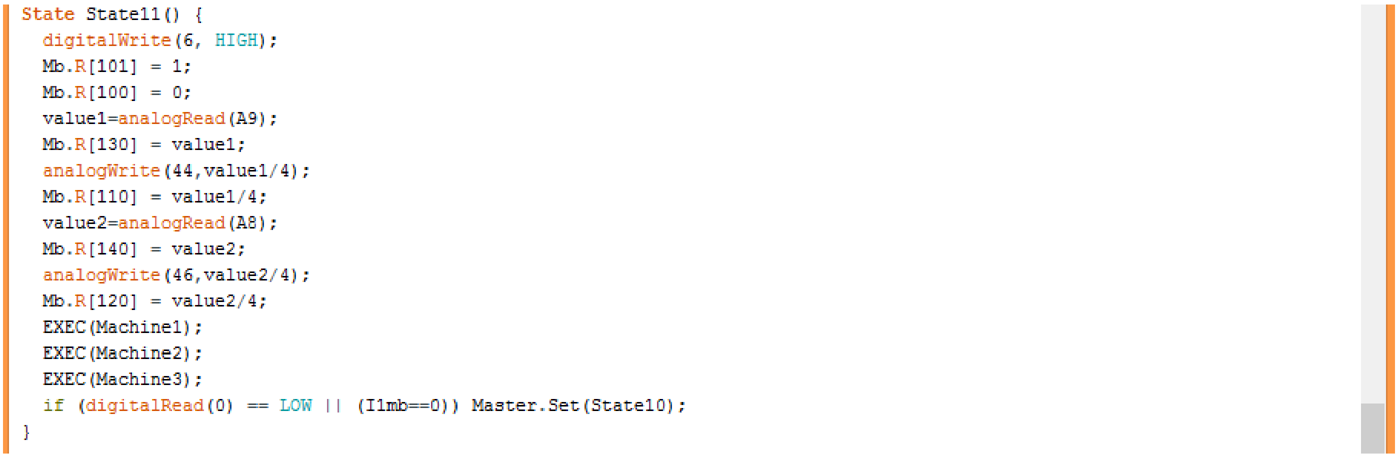
```
  DDRE = DDRE | B00001001; //D0 as output and then input, unless: not running
  DDRE = DDRE | B00001000;
  pinMode(14, INPUT);
  pinMode(9, INPUT);
  pinMode(8, INPUT);
  pinMode(15, INPUT);
  pinMode(16, INPUT);
  pinMode(17, INPUT);
  DDRD = DDRD | B00001111; //d18 d19 d20 d21 as output and then input, unless: not running
  DDRD = DDRD | B00000000;
  //pinMode(18, INPUT);//not needed!!!!!
  //pinMode(19, INPUT);
  //pinMode(20, INPUT);
  //pinMode(21, INPUT);
  pinMode(A15, INPUT);
  pinMode(A2, INPUT);
  pinMode(A3, INPUT);
  pinMode(A4, INPUT);
  pinMode(A5, INPUT);
  pinMode(A6, INPUT);
  //Mb.R[20] = 1500;

}
```



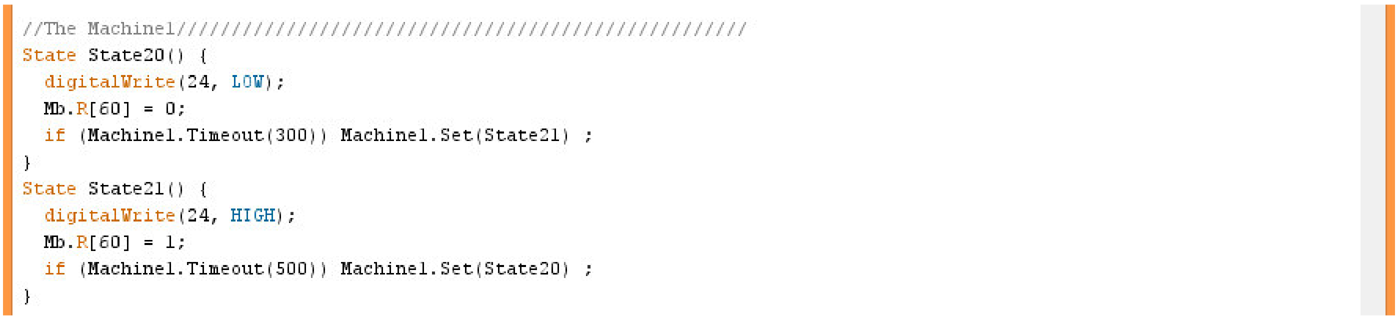
```
void loop() {
  Mb.Run();
   Mb.R[400] = 0;//hide a Pilotlight4 on HMI
  /*Mb.R[20] -> digitalRead(0);I1
  Mb.R[30] -> digitalRead(1);I2
  Mb.R[40] -> digitalRead(18);I11
  Mb.R[50] -> digitalRead(A4);I18
  Mb.R[60] <- digitalWrite(24, LOW);O1
  Mb.R[70] <- digitalWrite(5, LOW);O2
  Mb.R[80] <- digitalWrite(28, LOW);O6
  Mb.R[90] <- digitalWrite(A7, LOW);O12
  Mb.R[100] <- digitalWrite(6, LOW);O3
  */
  EXEC(Master);
  I1mb=Mb.R[20];
  if (digitalRead(0) == LOW || I1mb==0) {
    Machine1.Finish(); Machine2.Finish(); Machine3.Finish();
  }

  if (((digitalRead(0) == HIGH) && (I1mb==1) ) && Machine1.Finished && Machine2.Finished && Machine3.Finished) {
    EXEC(Machine1);
    Machine1.Set(State20);
    EXEC(Machine2);
    Machine2.Set(State30);
    EXEC(Machine3);
    Machine3.Set(State40);
  }
}
```



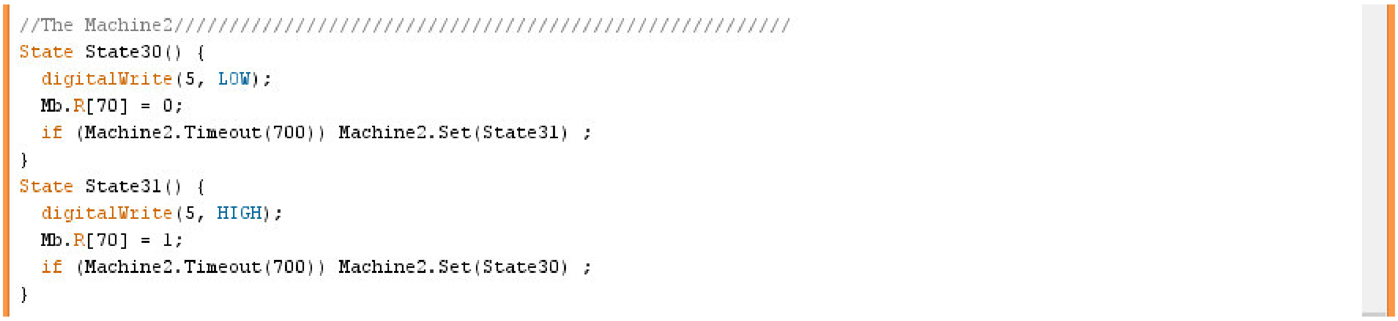
```
//The Master machine///////////////////////////////////////////////
State State10() {
  digitalWrite(24, LOW);
  Mb.R[60] = 0;
  digitalWrite(5, LOW);
  Mb.R[70] = 0;
  digitalWrite(28, LOW);
  Mb.R[80] = 0;
  digitalWrite(A7, LOW);
  Mb.R[90] = 0;
  digitalWrite(6, LOW);
  Mb.R[101] = 0;
  Mb.R[100] = 1;
  analogWrite(44,0);
  Mb.R[110] = 0;
  analogWrite(46,0);
  Mb.R[120] = 0;
  if ((digitalRead(0) == HIGH) && (I1mb==1)) Master.Set(State11);
}
```



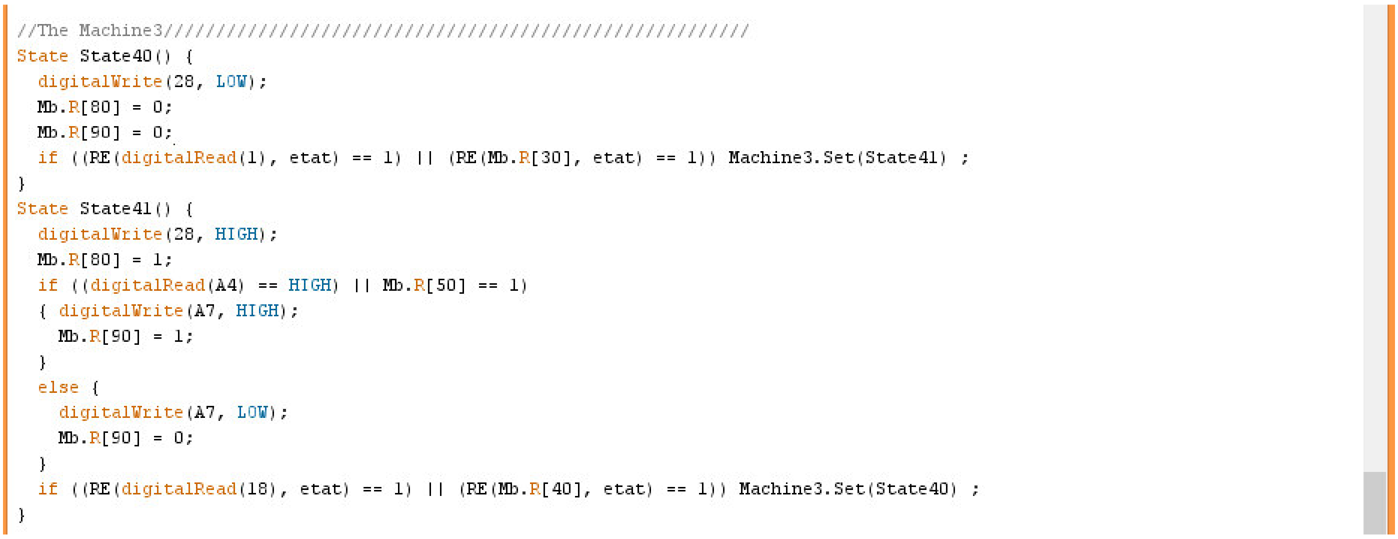
```
State State11() {
  digitalWrite(6, HIGH);
  Mb.R[101] = 1;
  Mb.R[100] = 0;
  value1=analogRead(A9);
  Mb.R[130] = value1;
  analogWrite(44,value1/4);
  Mb.R[110] = value1/4;
  value2=analogRead(A8);
  Mb.R[140] = value2;
  analogWrite(46,value2/4);
  Mb.R[120] = value2/4;
  EXEC(Machine1);
  EXEC(Machine2);
  EXEC(Machine3);
  if (digitalRead(0) == LOW || (I1mb==0)) Master.Set(State10);
}
```



```
//The Machine1///////////////////////////////////////////////////////
State State20() {
  digitalWrite(24, LOW);
  Mb.R[60] = 0;
  if (Machine1.Timeout(300)) Machine1.Set(State21) ;
}
State State21() {
  digitalWrite(24, HIGH);
  Mb.R[60] = 1;
  if (Machine1.Timeout(500)) Machine1.Set(State20) ;
}
```



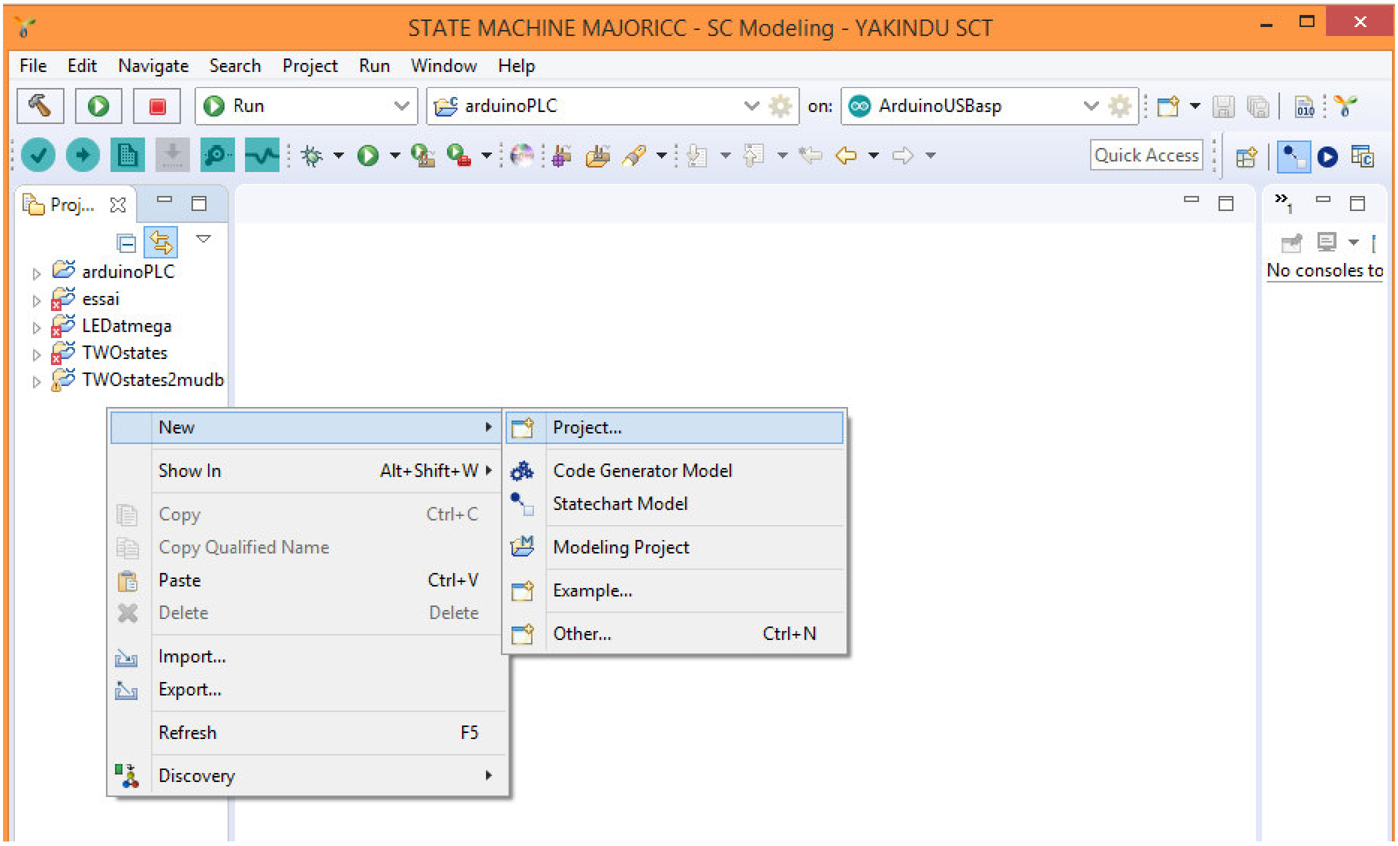
```
//The Machine2/////////////////////////////////////////////////////////////
State State30() {
  digitalWrite(5, LOW);
  Mb.R[70] = 0;
  if (Machine2.Timeout(700)) Machine2.Set(State31) ;
}
State State31() {
  digitalWrite(5, HIGH);
  Mb.R[70] = 1;
  if (Machine2.Timeout(700)) Machine2.Set(State30) ;
}
```

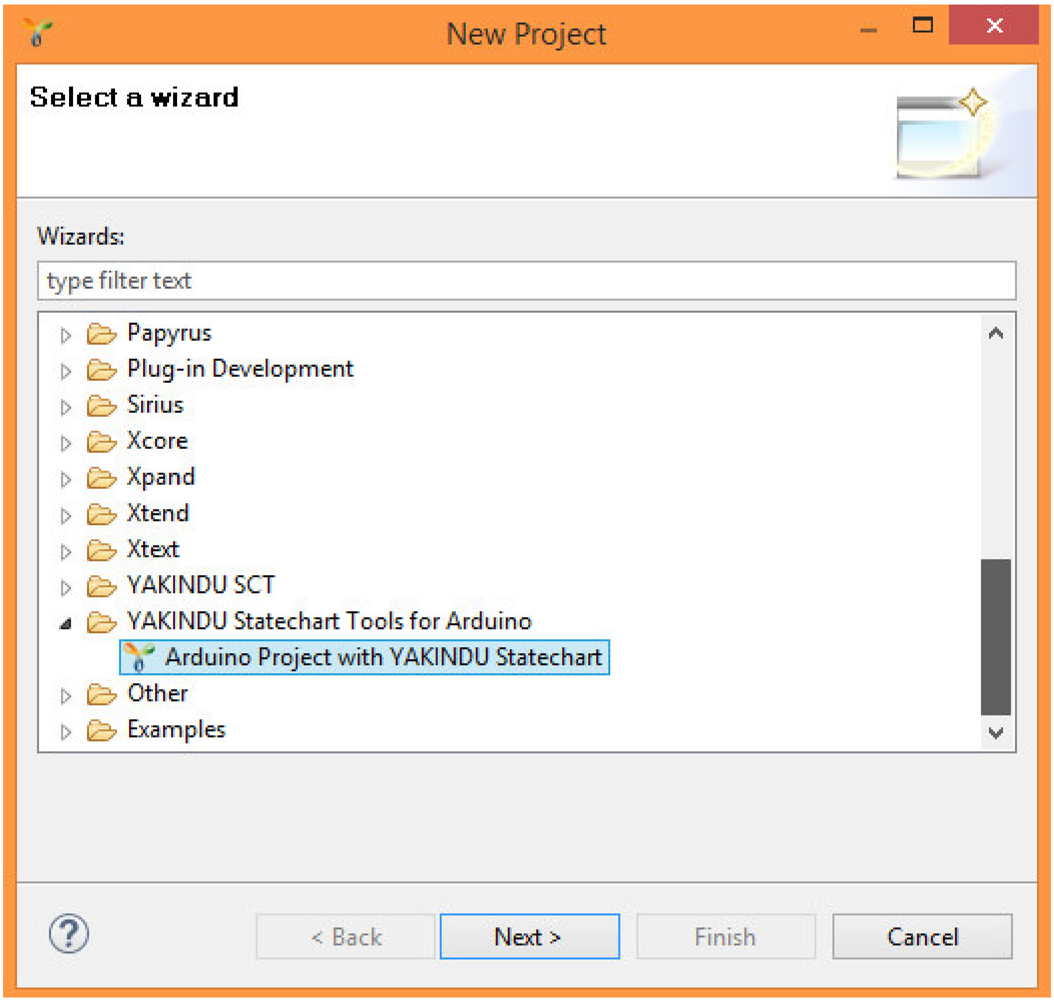


```
//The Machine3/////////////////////////////////////////////////////////////
State State40() {
  digitalWrite(28, LOW);
  Mb.R[80] = 0;
  Mb.R[90] = 0;
  if ((RE(digitalRead(1), etat) == 1) || (RE(Mb.R[30], etat) == 1)) Machine3.Set(State41) ;
}
State State41() {
  digitalWrite(28, HIGH);
  Mb.R[80] = 1;
  if ((digitalRead(A4) == HIGH) || Mb.R[50] == 1)
  { digitalWrite(A7, HIGH);
    Mb.R[90] = 1;
  }
  else {
    digitalWrite(A7, LOW);
    Mb.R[90] = 0;
  }
  if ((RE(digitalRead(18), etat) == 1) || (RE(Mb.R[40], etat) == 1)) Machine3.Set(State40) ;
}
```

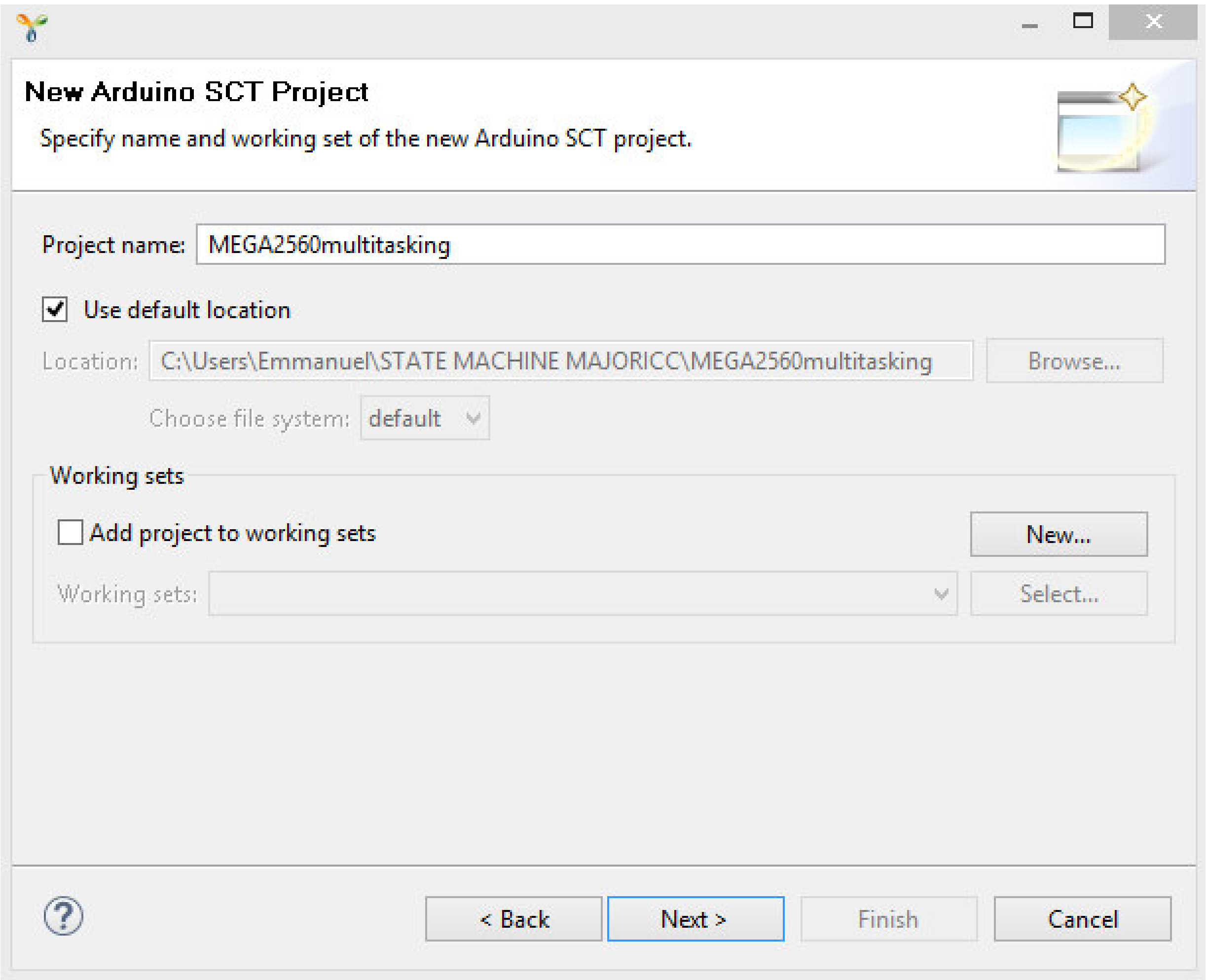
4-Multitasking with programming on Yakindu and using Arduino libraries inside:

4-1 Create the project:





Next



New Arduino SCT Project

Specify name and working set of the new Arduino SCT project.

Project name: MEGA2560multitasking

☑ Use default location

Location: C:\Users\Emmanuel\STATE MACHINE MAJORICC\MEGA2560multitasking

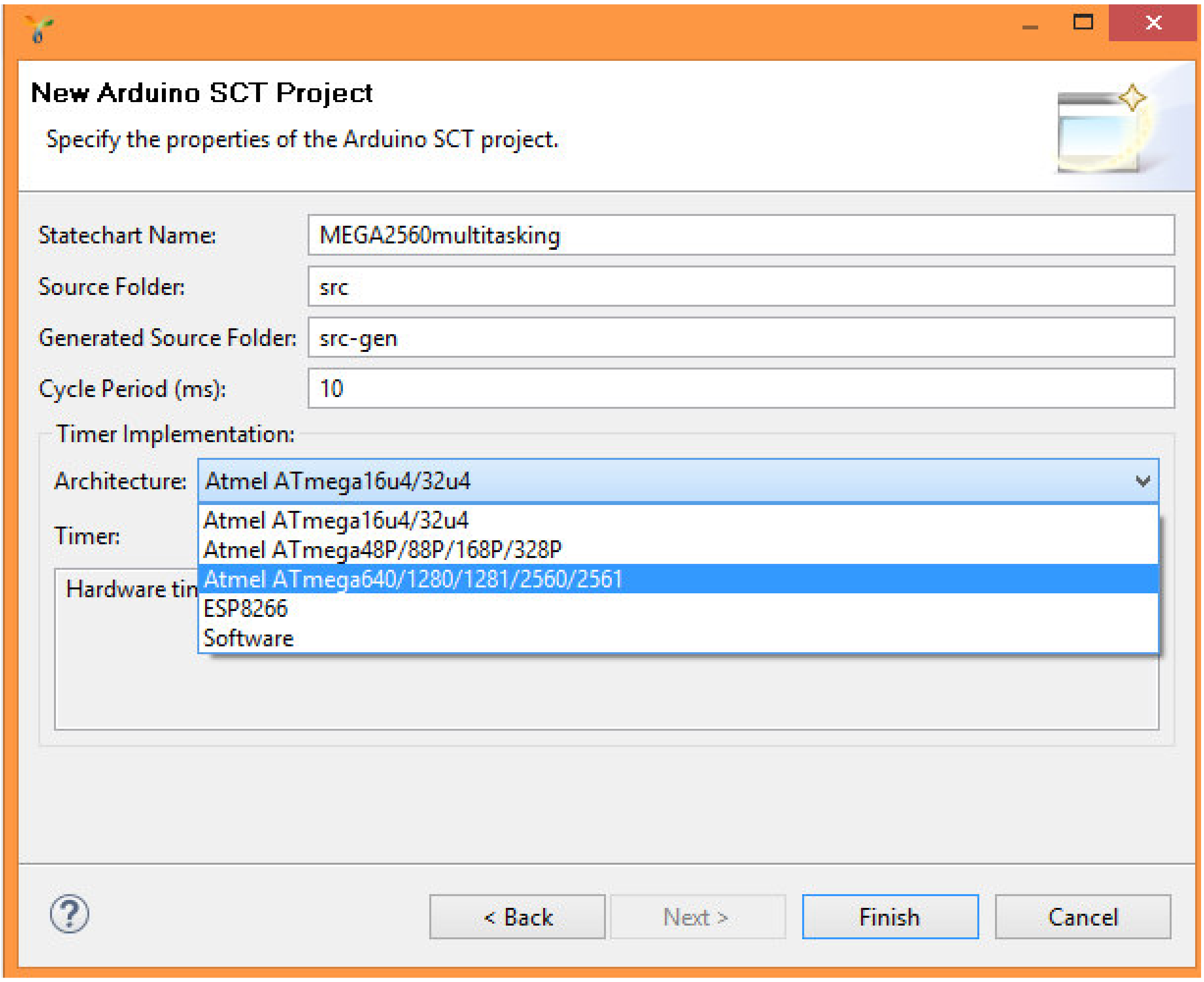
Choose file system: default

Working sets

□ Add project to working sets

Working sets:

Next



New Arduino SCT Project

Specify the properties of the Arduino SCT project.

Statechart Name: MEGA2560multitasking

Source Folder: src

Generated Source Folder: src-gen

Cycle Period (ms): 10

Timer Implementation:

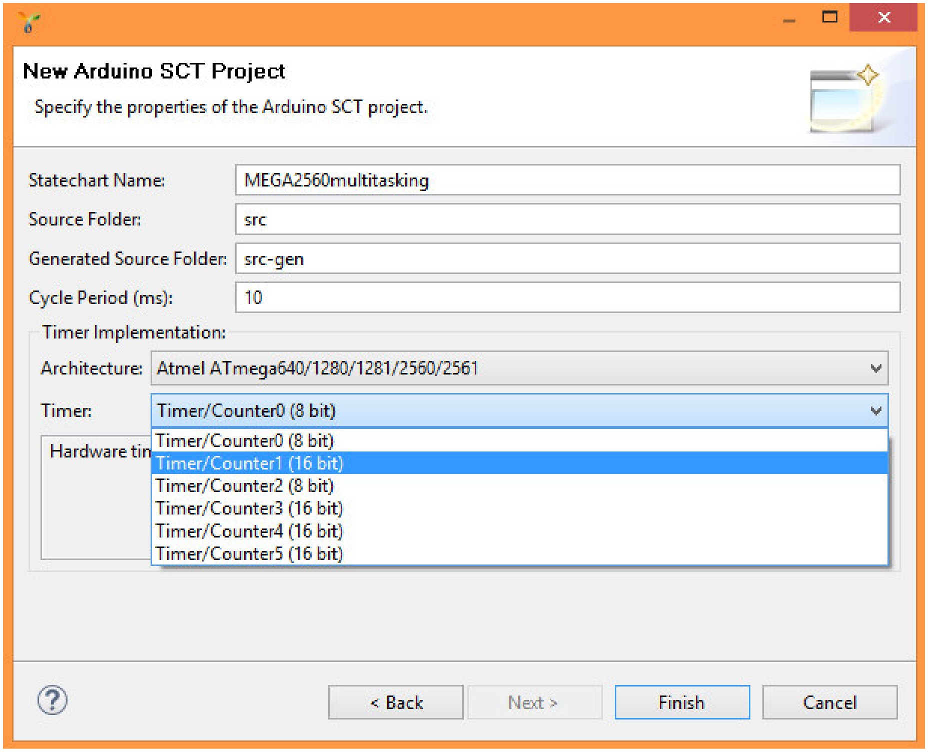
Architecture: Atmel ATmega16u4/32u4

Timer:

Hardware tin

Atmel ATmega16u4/32u4
Atmel ATmega48P/88P/168P/328P
Atmel ATmega640/1280/1281/2560/2561
ESP8266
Software
< Back
Next >
Finish
Cancel

Select the good Timer: Timer1 16 bit, if not you will have troubles to upload.



New Arduino SCT Project

Specify the properties of the Arduino SCT project.

Statechart Name: MEGA2560multitasking

Source Folder: src

Generated Source Folder: src-gen

Cycle Period (ms): 10

Timer Implementation:

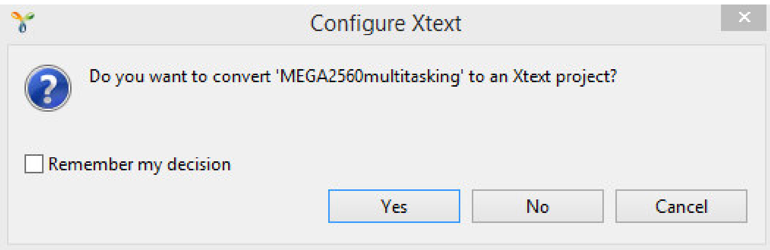
Architecture: Atmel ATmega640/1280/1281/2560/2561

Timer: Timer/Counter0 (8 bit)

- Timer/Counter0 (8 bit)
- Timer/Counter1 (16 bit)
- Timer/Counter2 (8 bit)
- Timer/Counter3 (16 bit)
- Timer/Counter4 (16 bit)
- Timer/Counter5 (16 bit)

Hardware tin

Finish



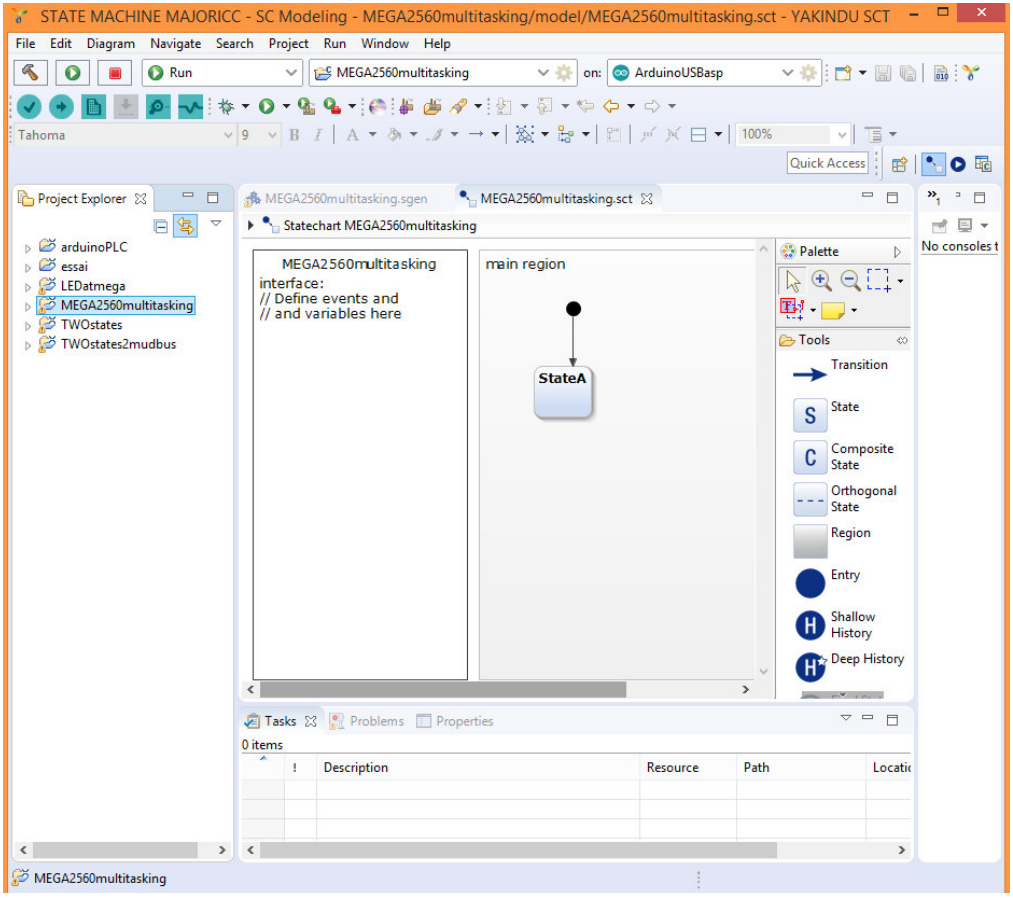
Configure Xtext

Do you want to convert 'MEGA2560multitasking' to an Xtext project?

□ Remember my decision

Yes

File MEGA2560multitasking Created:



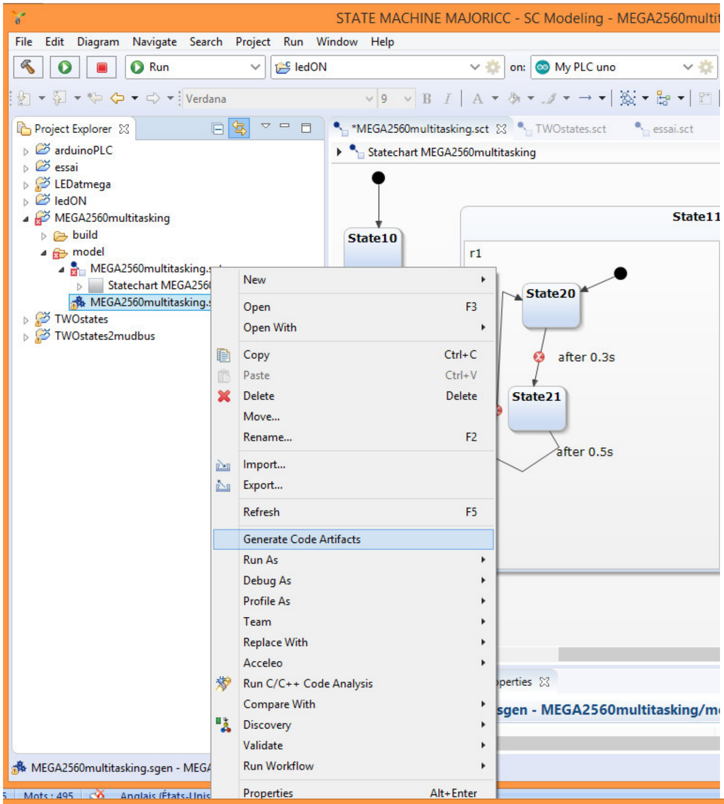
STATE MACHINE MAJORICC - SC Modeling - MEGA2560multitasking/model/MEGA2560multitasking.sct - YAKINDU SCT

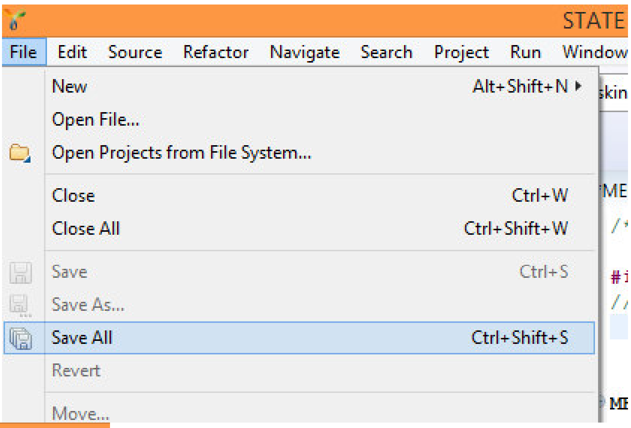
MEGA2560multitasking
interface:
// Define events and
// and variables here

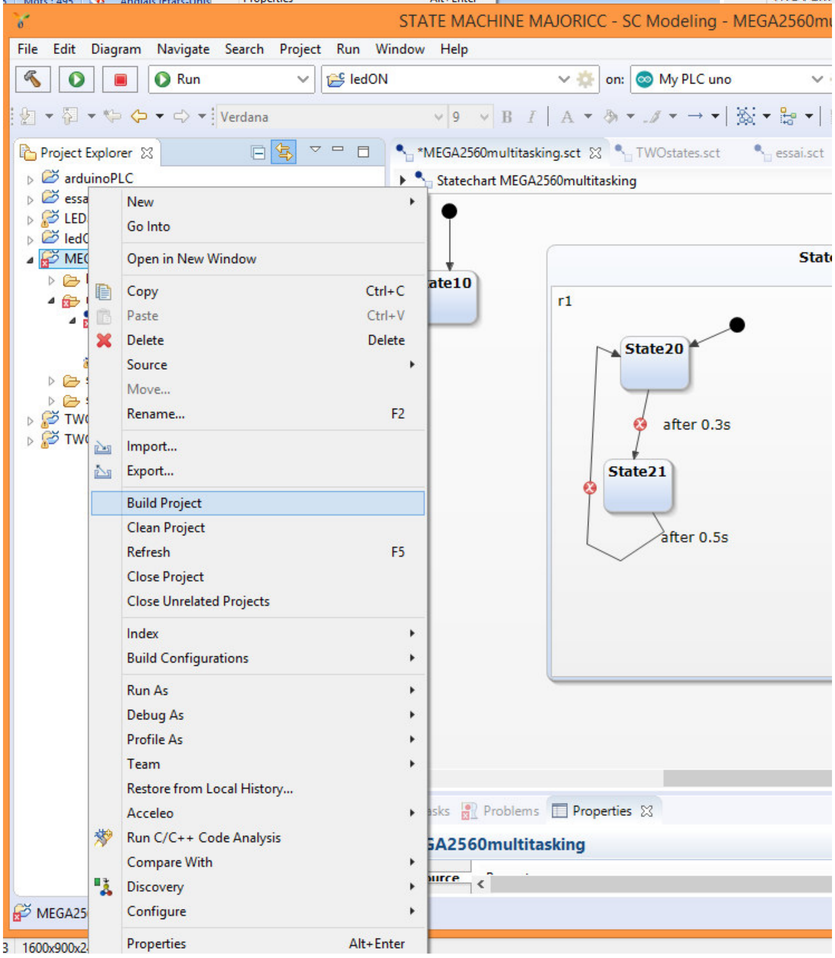
4-2 Drawings:

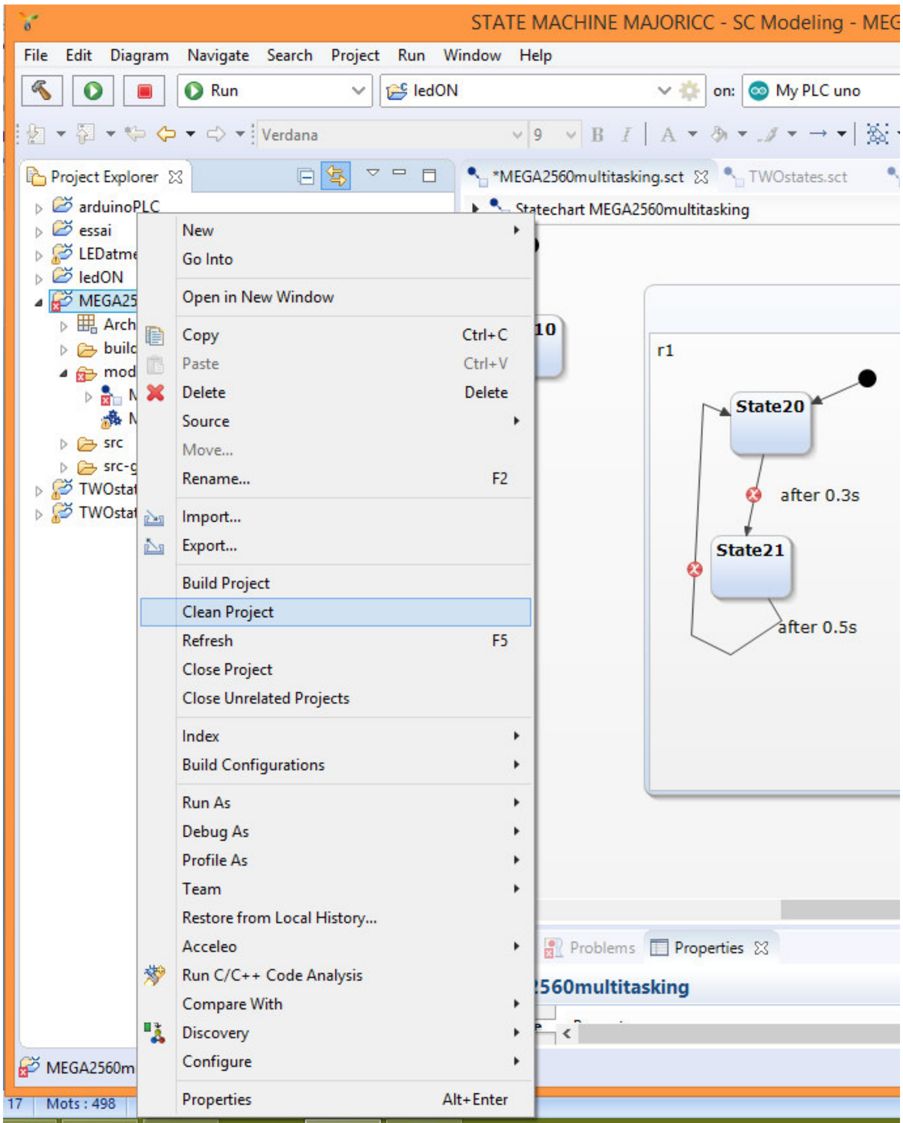
Now draw your state diagram:

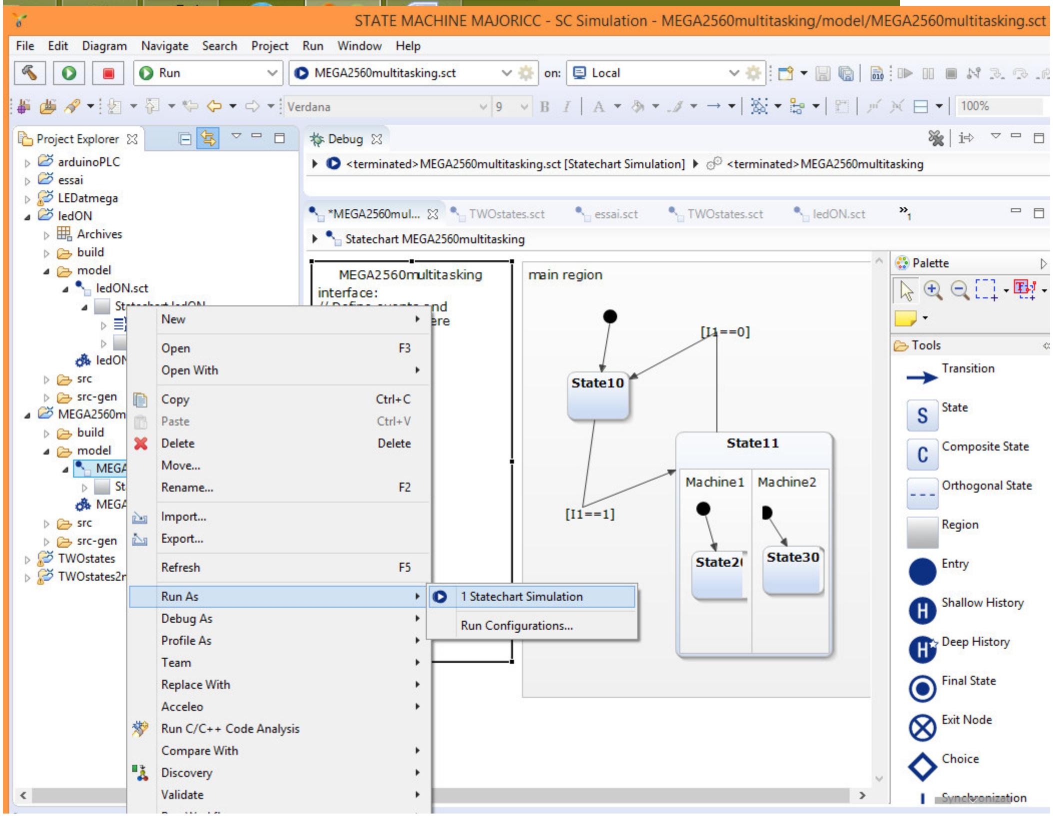
To prevent errors and test in local, some tricks to launch:





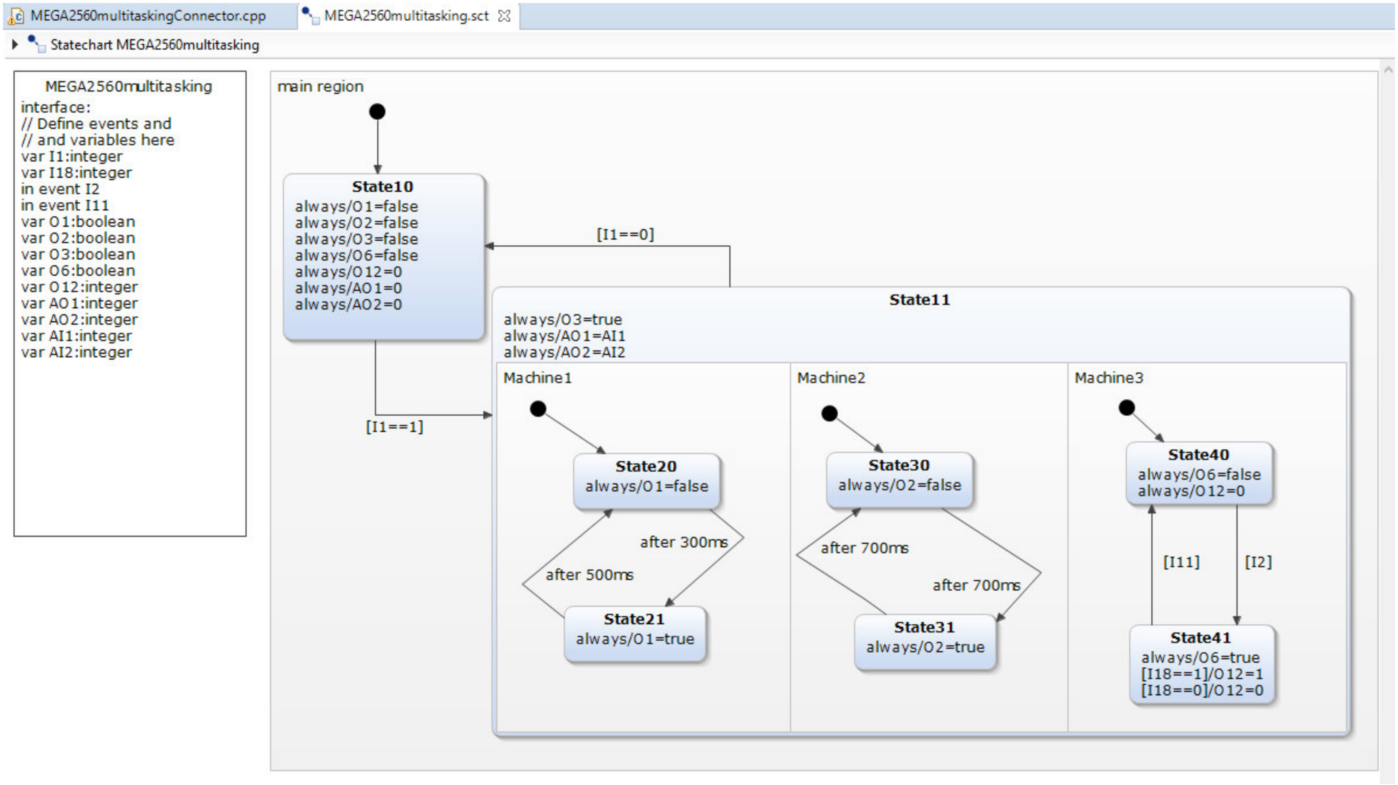






Sometimes you need to restart YAKINDU.

Draw the state diagram

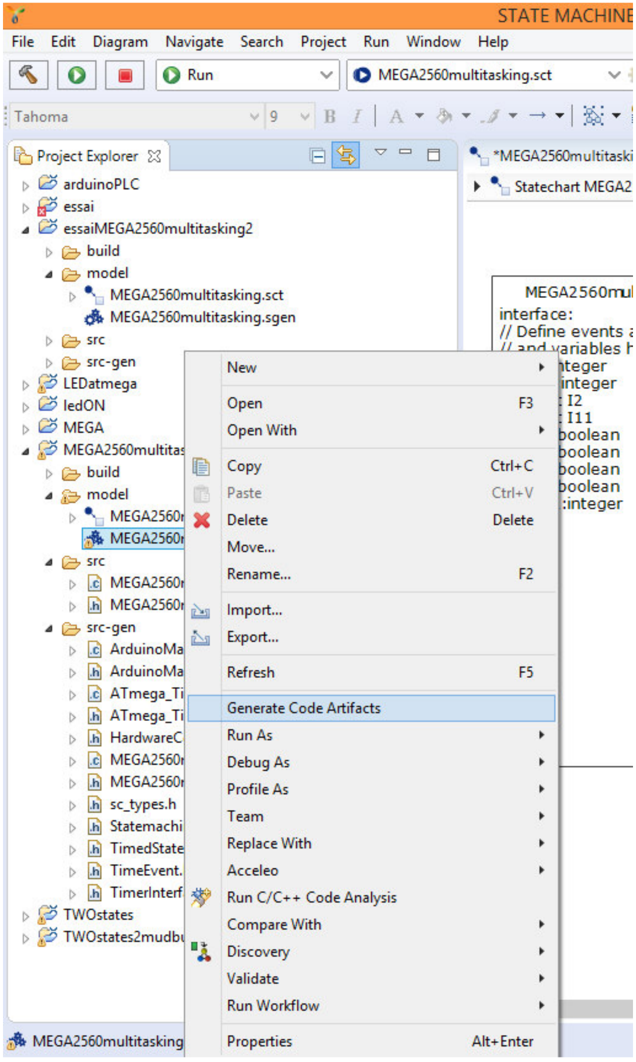


```
MEGA2560multitasking
interface:
// Define events and
// and variables here
var I1:integer
var I18:integer
in event I2
in event I11
var O1:boolean
var O2:boolean
var O3:boolean
var O6:boolean
var O12:integer
var AO1:integer
var AO2:integer
var AI1:integer
var AI2:integer
```

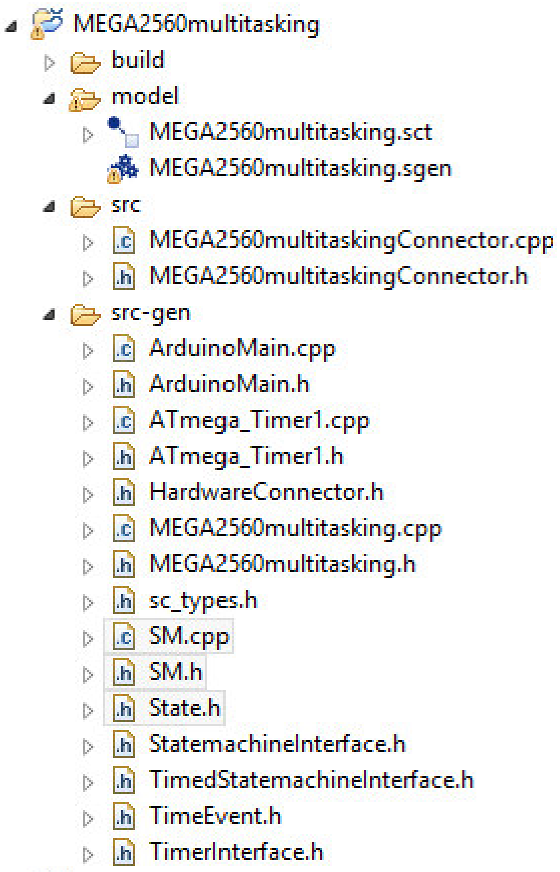
I1 declared as integer if you don't want a RE (rising edge) on the transition.
I2 declared as in event to use the RE function.

4-3 Generate the code:

Now generate the code:

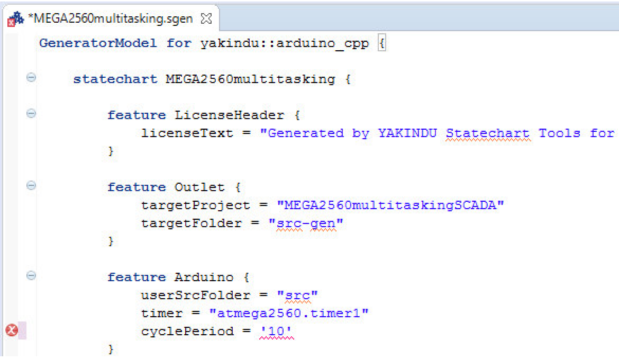


Import Arduino libraries needed: SMlib in this case because I need the RE function. Put it in src-gen



```
MEGA2560multitasking
    build
    model
        MEGA2560multitasking.sct
        MEGA2560multitasking.sgen
    src
        MEGA2560multitaskingConnector.cpp
        MEGA2560multitaskingConnector.h
    src-gen
        ArduinoMain.cpp
        ArduinoMain.h
        ATmega_Timer1.cpp
        ATmega_Timer1.h
        HardwareConnector.h
        MEGA2560multitasking.cpp
        MEGA2560multitasking.h
        sc_types.h
        SM.cpp
        SM.h
        State.h
        StatemachineInterface.h
        TimedStatemachineInterface.h
        TimeEvent.h
        TimerInterface.h
```

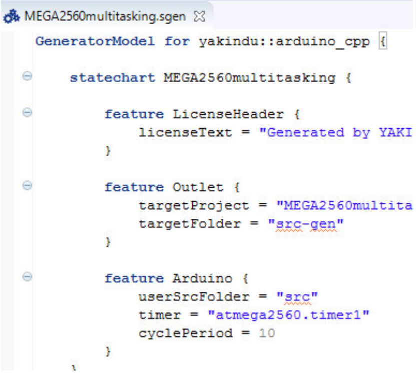
Sometimes with Yakindu update the file xxx.sgen has an error:



*MEGA2560multitasking.sgen

```
GeneratorModel for yakindu::arduino_cpp {

    statechart MEGA2560multitasking {

        feature LicenseHeader {
            licenseText = "Generated by YAKINDU Statechart Tools for
        }

        feature Outlet {
            targetProject = "MEGA2560multitaskingSCADA"
            targetFolder = "src-gen"
        }

        feature Arduino {
            userSrcFolder = "src"
            timer = "atmega2560.timer1"
            cyclePeriod = '10'
        }
```

You just have to remove the quote of the cyclPeriod:



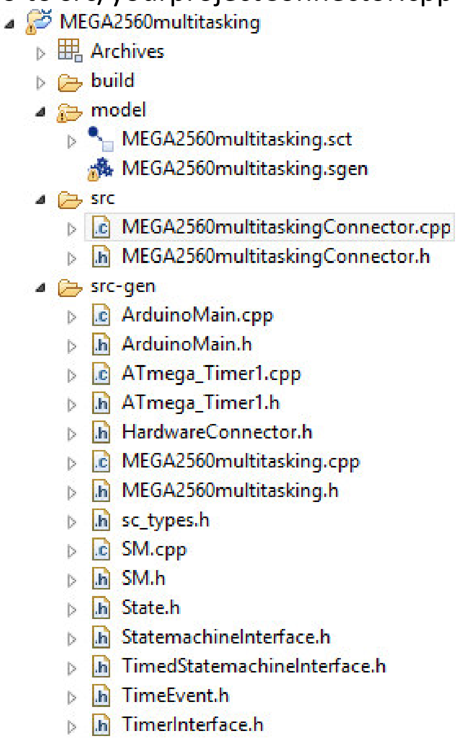
MEGA2560multitasking.sgen

```
GeneratorModel for yakindu::arduino_cpp {

    statechart MEGA2560multitasking {

        feature LicenseHeader {
            licenseText = "Generated by YAKI
        }

        feature Outlet {
            targetProject = "MEGA2560multita
            targetFolder = "src-gen"
        }

        feature Arduino {
            userSrcFolder = "src"
            timer = "atmega2560.timer1"
            cyclePeriod = 10
        }
```

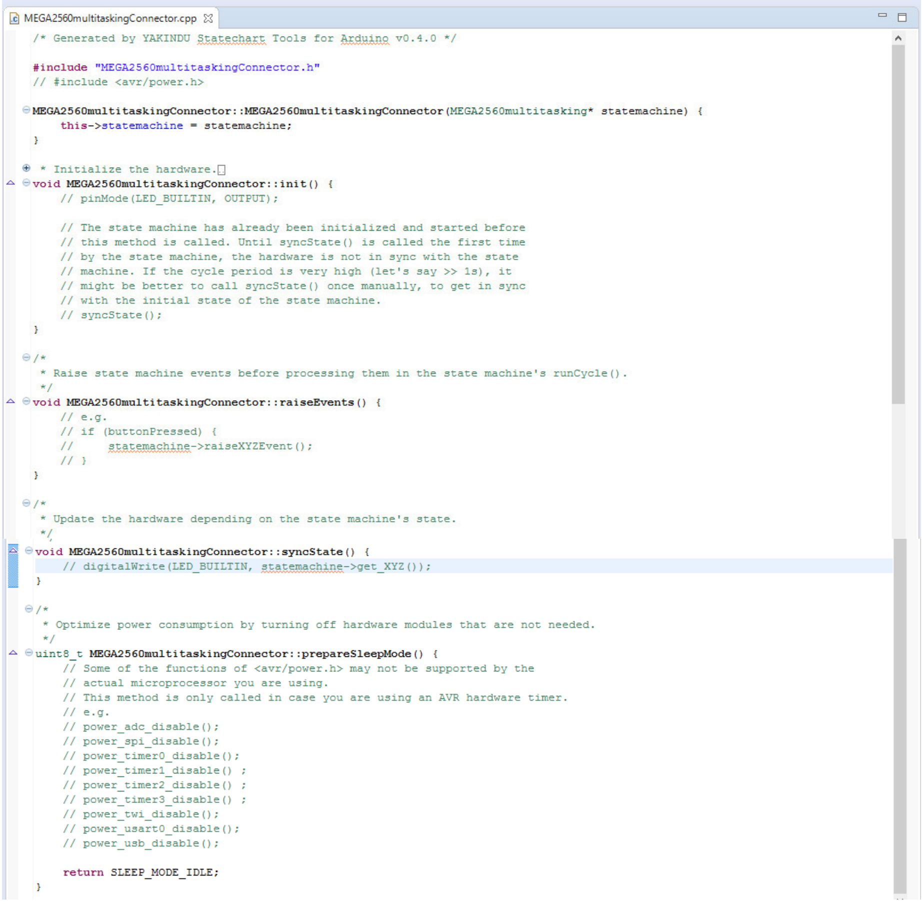
Build and clean the project.

4-4 Complete the code:

Go to src/yourprojectConnector.cpp to modify the code to upload:



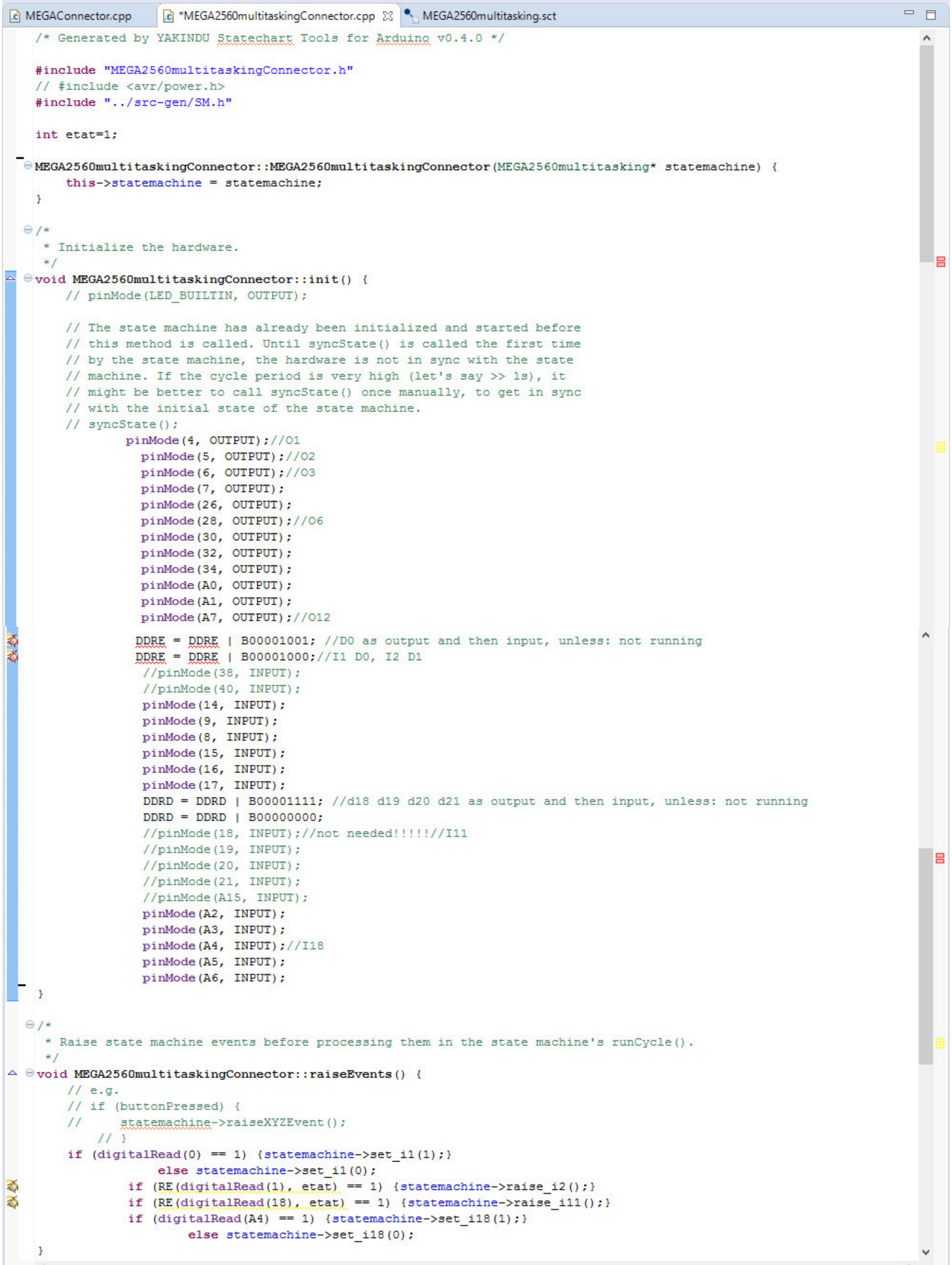
```
MEGA2560multitasking
    Archives
    build
    model
        MEGA2560multitasking.sct
        MEGA2560multitasking.sgen
    src
        MEGA2560multitaskingConnector.cpp
        MEGA2560multitaskingConnector.h
    src-gen
        ArduinoMain.cpp
        ArduinoMain.h
        ATmega_Timer1.cpp
        ATmega_Timer1.h
        HardwareConnector.h
        MEGA2560multitasking.cpp
        MEGA2560multitasking.h
        sc_types.h
        SM.cpp
        SM.h
        State.h
        StatemachineInterface.h
        TimedStatemachineInterface.h
        TimeEvent.h
        TimerInterface.h
```



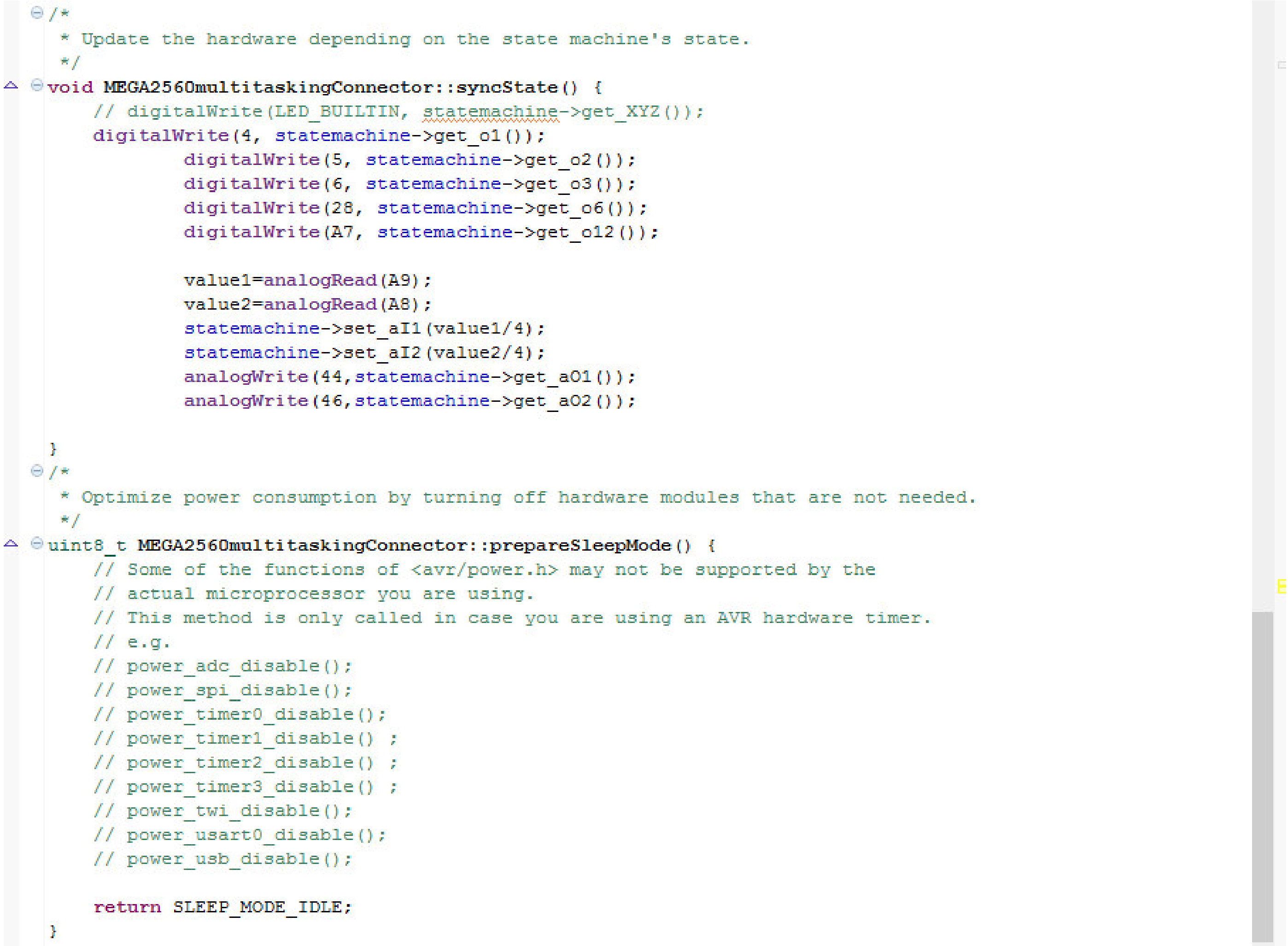
MEGA2560multitaskingConnector.cpp

```
/* Generated by YAKINDU Statechart Tools for Arduino v0.4.0 */

#include "MEGA2560multitaskingConnector.h"
// #include <avr/power.h>

MEGA2560multitaskingConnector::MEGA2560multitaskingConnector(MEGA2560multitasking* statemachine) {
    this->statemachine = statemachine;
}

 * Initialize the hardware.
void MEGA2560multitaskingConnector::init() {
    // pinMode(LED_BUILTIN, OUTPUT);

    // The state machine has already been initialized and started before
    // this method is called. Until syncState() is called the first time
    // by the state machine, the hardware is not in sync with the state
    // machine. If the cycle period is very high (let's say >> 1s), it
    // might be better to call syncState() once manually, to get in sync
    // with the initial state of the state machine.
    // syncState();
}

/*
 * Raise state machine events before processing them in the state machine's runCycle().
 */
void MEGA2560multitaskingConnector::raiseEvents() {
    // e.g.
    // if (buttonPressed) {
    //     statemachine->raiseXYZEvent();
    // }
}

/*
 * Update the hardware depending on the state machine's state.
 */
void MEGA2560multitaskingConnector::syncState() {
    // digitalWrite(LED_BUILTIN, statemachine->get_XYZ());
}

/*
 * Optimize power consumption by turning off hardware modules that are not needed.
 */
uint8_t MEGA2560multitaskingConnector::prepareSleepMode() {
    // Some of the functions of <avr/power.h> may not be supported by the
    // actual microprocessor you are using.
    // This method is only called in case you are using an AVR hardware timer.
    // e.g.
    // power_adc_disable();
    // power_spi_disable();
    // power_timer0_disable();
    // power_timer1_disable() ;
    // power_timer2_disable() ;
    // power_timer3_disable() ;
    // power_twi_disable();
    // power_usart0_disable();
    // power_usb_disable();

    return SLEEP_MODE_IDLE;
}
```

The modified and completed file:



```
/* Generated by YAKINDU Statechart Tools for Arduino v0.4.0 */

#include "MEGA2560multitaskingConnector.h"
// #include <avr/power.h>
#include "../src-gen/SM.h"

int etat=1;

MEGA2560multitaskingConnector::MEGA2560multitaskingConnector(MEGA2560multitasking* statemachine) {
    this->statemachine = statemachine;
}

/*
 * Initialize the hardware.
 */
void MEGA2560multitaskingConnector::init() {
    // pinMode(LED_BUILTIN, OUTPUT);

    // The state machine has already been initialized and started before
    // this method is called. Until syncState() is called the first time
    // by the state machine, the hardware is not in sync with the state
    // machine. If the cycle period is very high (let's say >> 1s), it
    // might be better to call syncState() once manually, to get in sync
    // with the initial state of the state machine.
    // syncState();
            pinMode(4, OUTPUT);//O1
              pinMode(5, OUTPUT);//O2
              pinMode(6, OUTPUT);//O3
              pinMode(7, OUTPUT);
              pinMode(26, OUTPUT);
              pinMode(28, OUTPUT);//O6
              pinMode(30, OUTPUT);
              pinMode(32, OUTPUT);
              pinMode(34, OUTPUT);
              pinMode(A0, OUTPUT);
              pinMode(A1, OUTPUT);
              pinMode(A7, OUTPUT);//O12

             DDRE = DDRE | B00001001; //D0 as output and then input, unless: not running
             DDRE = DDRE | B00001000;//I1 D0, I2 D1
              //pinMode(38, INPUT);
              //pinMode(40, INPUT);
              pinMode(14, INPUT);
              pinMode(9, INPUT);
              pinMode(8, INPUT);
              pinMode(15, INPUT);
              pinMode(16, INPUT);
              pinMode(17, INPUT);
              DDRD = DDRD | B00001111; //d18 d19 d20 d21 as output and then input, unless: not running
              DDRD = DDRD | B00000000;
              //pinMode(18, INPUT);//not needed!!!!!//I11
              //pinMode(19, INPUT);
              //pinMode(20, INPUT);
              //pinMode(21, INPUT);
              //pinMode(A15, INPUT);
              pinMode(A2, INPUT);
              pinMode(A3, INPUT);
              pinMode(A4, INPUT);//I18
              pinMode(A5, INPUT);
              pinMode(A6, INPUT);
}

/*
 * Raise state machine events before processing them in the state machine's runCycle().
 */
void MEGA2560multitaskingConnector::raiseEvents() {
    // e.g.
    // if (buttonPressed) {
    //       statemachine->raiseXYZEvent();
        // }
    if (digitalRead(0) == 1) {statemachine->set_i1(1);}
                else statemachine->set_i1(0);
            if (RE(digitalRead(1), etat) == 1) {statemachine->raise_i2();}
            if (RE(digitalRead(18), etat) == 1) {statemachine->raise_i11();}
            if (digitalRead(A4) == 1) {statemachine->set_i18(1);}
                    else statemachine->set_i18(0);
}
```

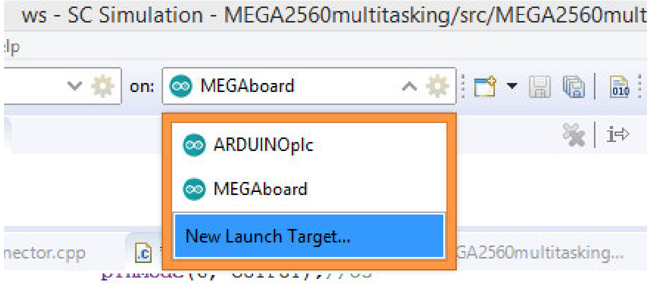


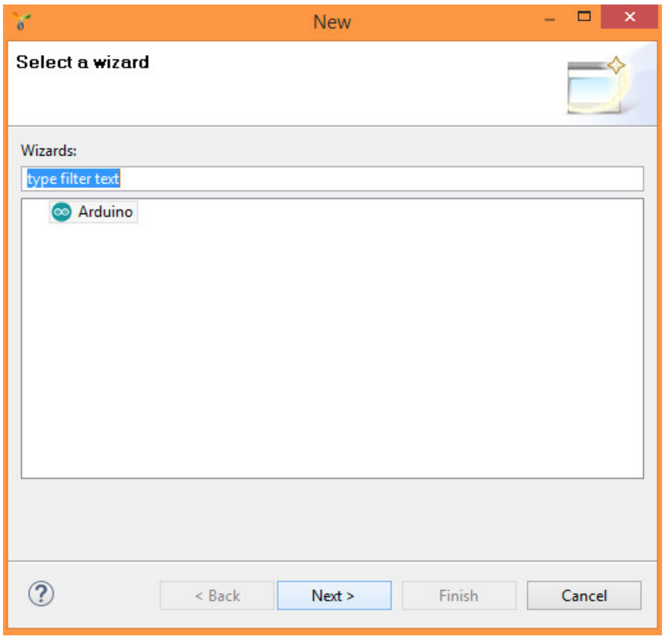
```
/*
 * Update the hardware depending on the state machine's state.
 */
void MEGA2560multitaskingConnector::syncState() {
    // digitalWrite(LED_BUILTIN, statemachine->get_XYZ());
    digitalWrite(4, statemachine->get_o1());
            digitalWrite(5, statemachine->get_o2());
            digitalWrite(6, statemachine->get_o3());
            digitalWrite(28, statemachine->get_o6());
            digitalWrite(A7, statemachine->get_o12());

            value1=analogRead(A9);
            value2=analogRead(A8);
            statemachine->set_aI1(value1/4);
            statemachine->set_aI2(value2/4);
            analogWrite(44,statemachine->get_aO1());
            analogWrite(46,statemachine->get_aO2());

}
/*
 * Optimize power consumption by turning off hardware modules that are not needed.
 */
uint8_t MEGA2560multitaskingConnector::prepareSleepMode() {
    // Some of the functions of <avr/power.h> may not be supported by the
    // actual microprocessor you are using.
    // This method is only called in case you are using an AVR hardware timer.
    // e.g.
    // power_adc_disable();
    // power_spi_disable();
    // power_timer0_disable();
    // power_timer1_disable() ;
    // power_timer2_disable() ;
    // power_timer3_disable() ;
    // power_twi_disable();
    // power_usart0_disable();
    // power_usb_disable();

    return SLEEP_MODE_IDLE;
}
```

init(): a part to declare Inputs and output. It's like Setup().
raiseEvents(): A part for real inputs/transitions links
syncState(): a part for outputs/actions links. Sometimes the functions like "`statemachine->get_o1()`" or else give errors, so you need to clean and rebuild the profect until it appears in src-gen/MEGA2560multitasking.h file.
Build, clean and here remains an error on DDRE (I don't know why but it works) and you can still upload the program.

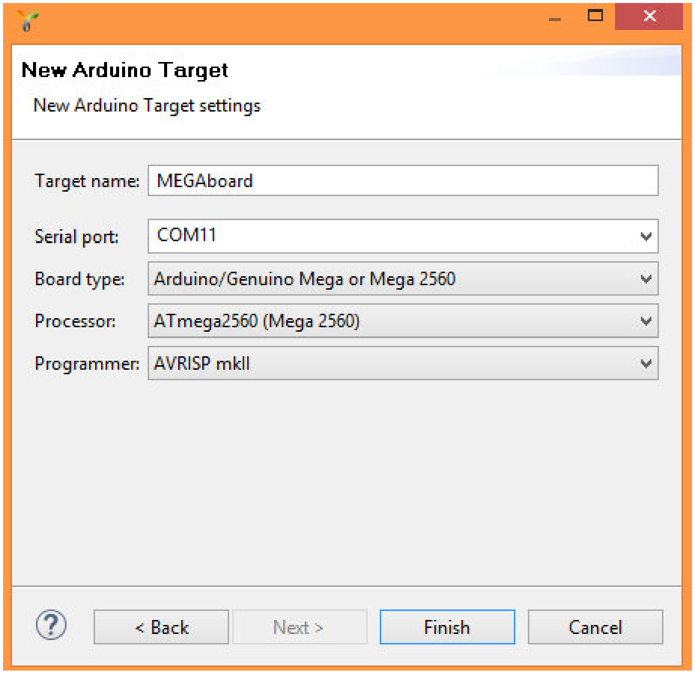
4-5 Upload the code:

Before uploading: create a driver





Next, complete:



Target name: MEGAboard

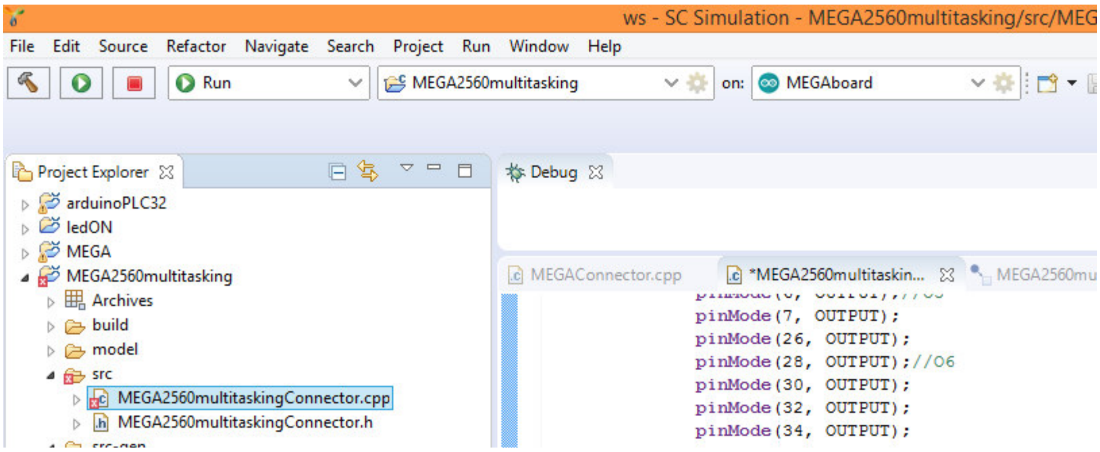
Serial port: COM11

Board type: Arduino/Genuino Mega or Mega 2560

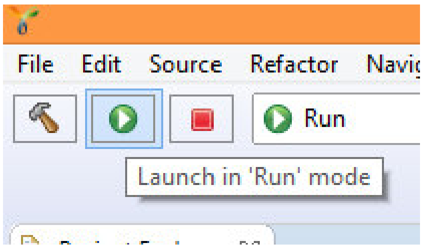
Processor: ATmega2560 (Mega 2560)

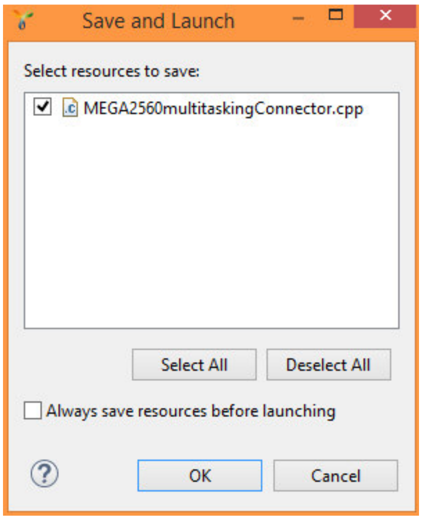
Programmer: AVRISP mkII

Finish



```
pinMode(7, OUTPUT);
pinMode(26, OUTPUT);
pinMode(28, OUTPUT);//O6
pinMode(30, OUTPUT);
pinMode(32, OUTPUT);
pinMode(34, OUTPUT);
```



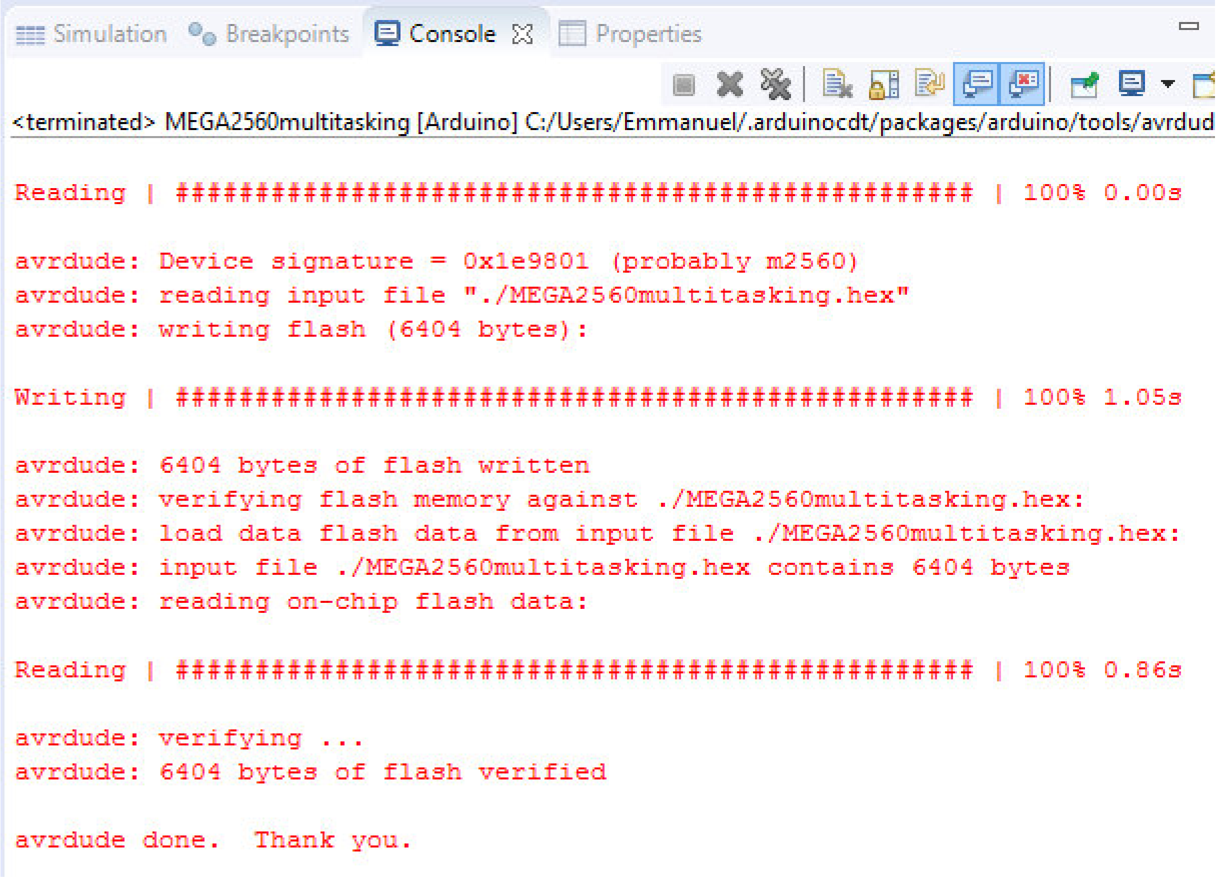


Select resources to save:

☑ MEGA2560multitaskingConnector.cpp

□ Always save resources before launching

The result, if it’s good:

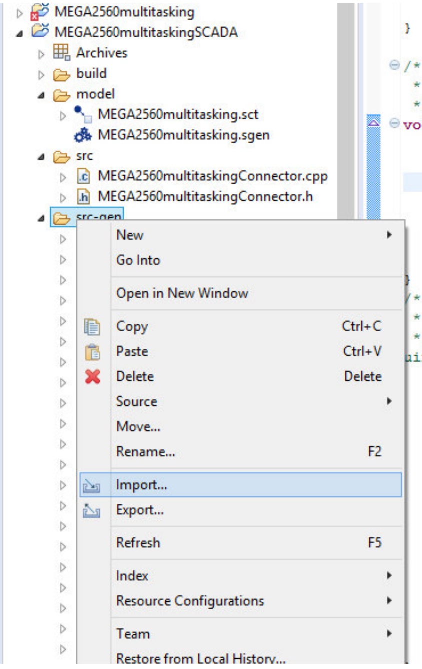


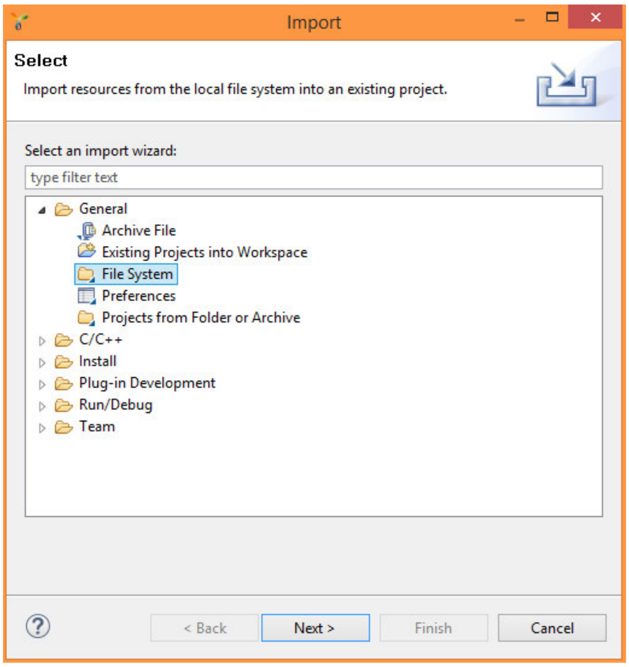
```
<terminated> MEGA2560multitasking [Arduino] C:/Users/Emmanuel/.arduinocdt/packages/arduino/tools/avrdud

Reading | ################################################## | 100% 0.00s

avrdude: Device signature = 0x1e9801 (probably m2560)
avrdude: reading input file "./MEGA2560multitasking.hex"
avrdude: writing flash (6404 bytes):

Writing | ################################################## | 100% 1.05s

avrdude: 6404 bytes of flash written
avrdude: verifying flash memory against ./MEGA2560multitasking.hex:
avrdude: load data flash data from input file ./MEGA2560multitasking.hex:
avrdude: input file ./MEGA2560multitasking.hex contains 6404 bytes
avrdude: reading on-chip flash data:

Reading | ################################################## | 100% 0.86s

avrdude: verifying ...
avrdude: 6404 bytes of flash verified

avrdude done.  Thank you.
```

5-Multitasking with programming on Yakindu using Arduino libraries inside and supervising on advancedHMI:

5-1 Import needed libraries:

You need to import some libraries into your src-gen file in YAKINDU:

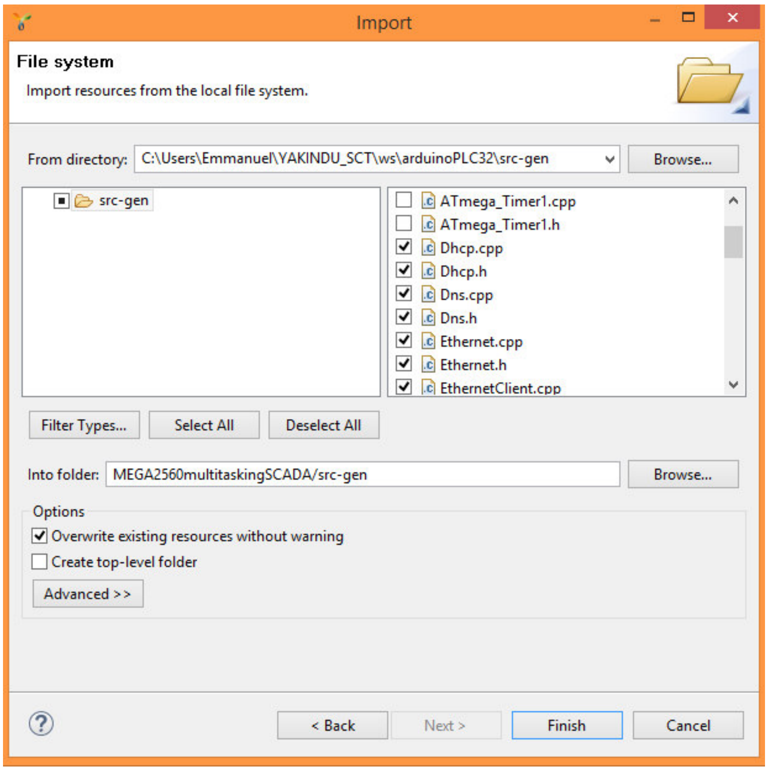




Select

Import resources from the local file system into an existing project.

Select an import wizard:



File system

Import resources from the local file system.

From directory: C:\Users\Emmanuel\YAKINDU_SCT\ws\arduinoPLC32\src-gen

■ src-gen

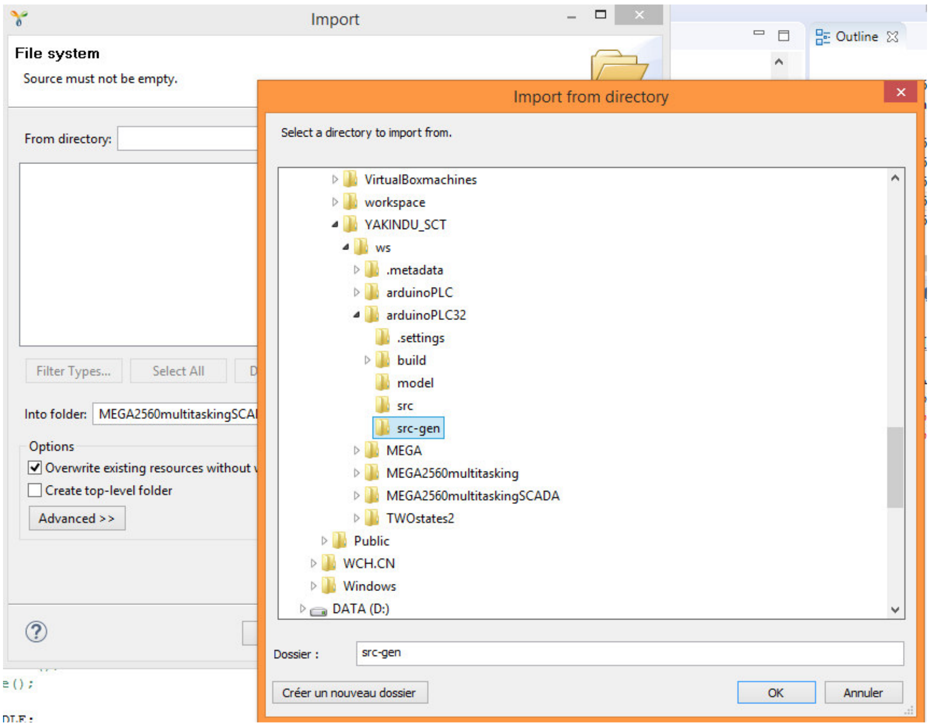
- ☐ ATmega_Timer1.cpp
- ☐ ATmega_Timer1.h
- ☑ Dhcp.cpp
- ☑ Dhcp.h
- ☑ Dns.cpp
- ☑ Dns.h
- ☑ Ethernet.cpp
- ☑ Ethernet.h
- ☑ EthernetClient.cpp

Into folder: MEGA2560multitaskingSCADA/src-gen

Options

☑ Overwrite existing resources without warning

☐ Create top-level folder



Import

File system

Source must not be empty.

From directory:

Into folder: MEGA2560multitaskingSCAI

Options

☑ Overwrite existing resources without w

□ Create top-level folder

Import from directory

Select a directory to import from.

Dossier : src-gen

From a previous src-gen file project:

If there are troubles in building with some imported library: correct **#include** `<myLib>` by **#include** `“myLib”`

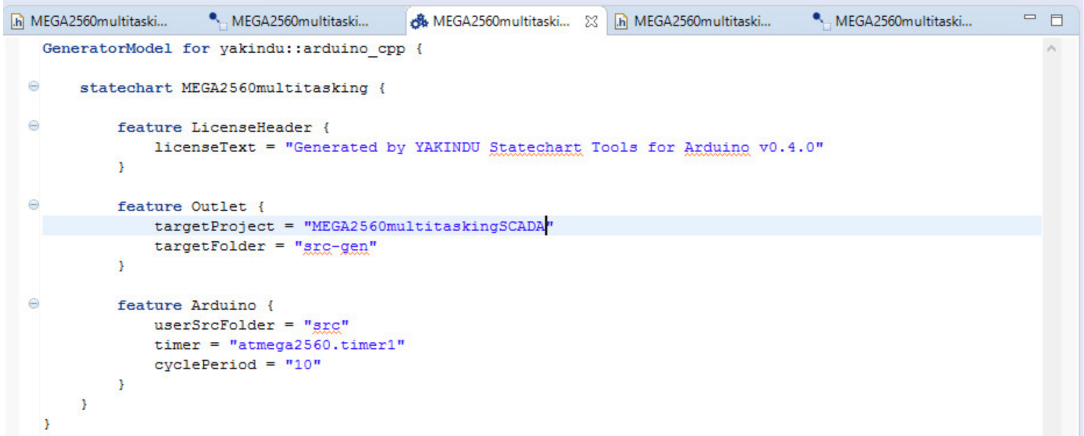
If a library includes folders, put all the files in the same YAKINDU folder: src-gen and don’t forget: “save/clean/build” to correct each error.

To insert library put it in the src-gen folder or right click on src-gen/import files and select the folder where is your library to add. Then call it in src/xxxConnector.cpp:

#include `"../src-gen/thelib.h"`

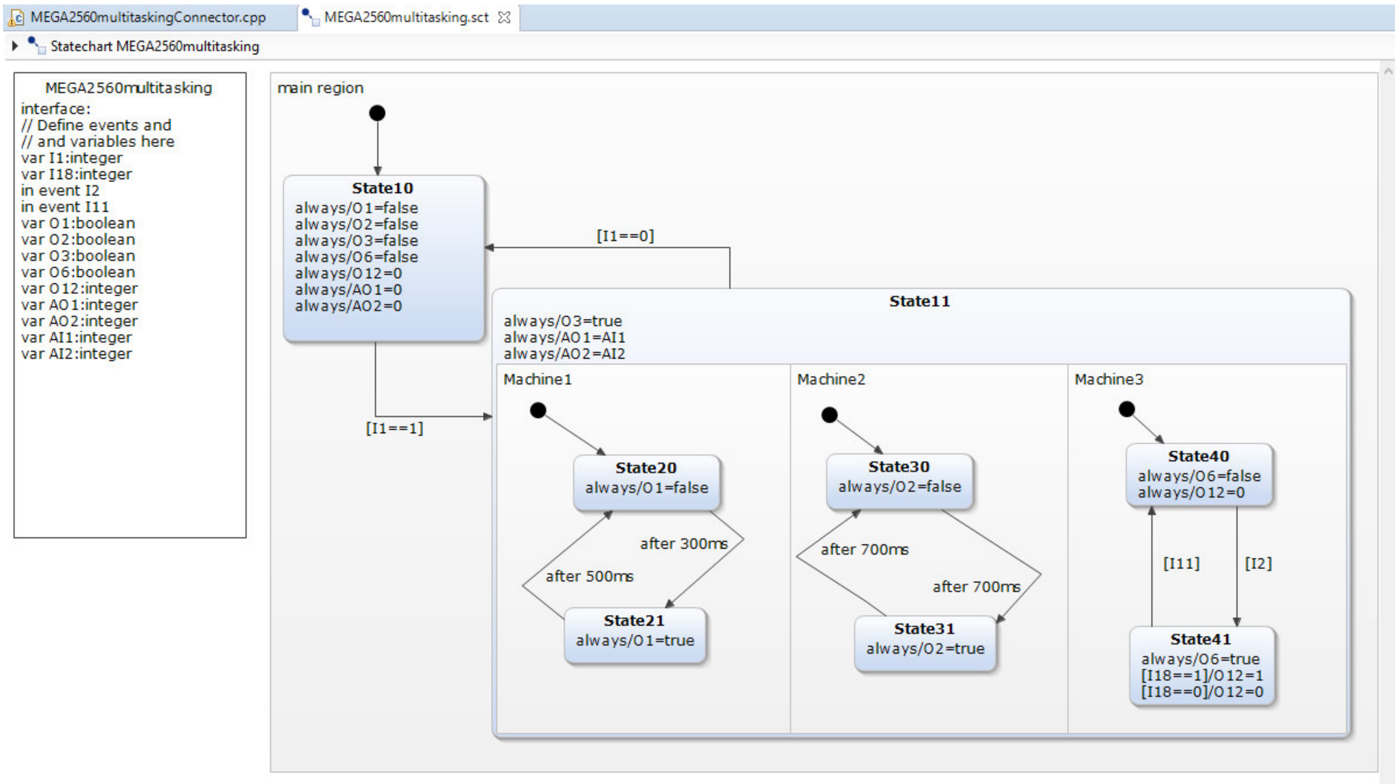
5-2 Copy the previous project

If you work on a copy of a project, becarefull to change the target where the code is generated:



```
GeneratorModel for yakindu::arduino_cpp {

    statechart MEGA2560multitasking {

        feature LicenseHeader {
            licenseText = "Generated by YAKINDU Statechart Tools for Arduino v0.4.0"
        }

        feature Outlet {
            targetProject = "MEGA2560multitaskingSCADA"
            targetFolder = "src-gen"
        }

        feature Arduino {
            userSrcFolder = "src"
            timer = "atmega2560.timer1"
            cyclePeriod = "10"
        }
    }
}
```

Some changes compared with the previous state diagram: add virtual HMI buttons

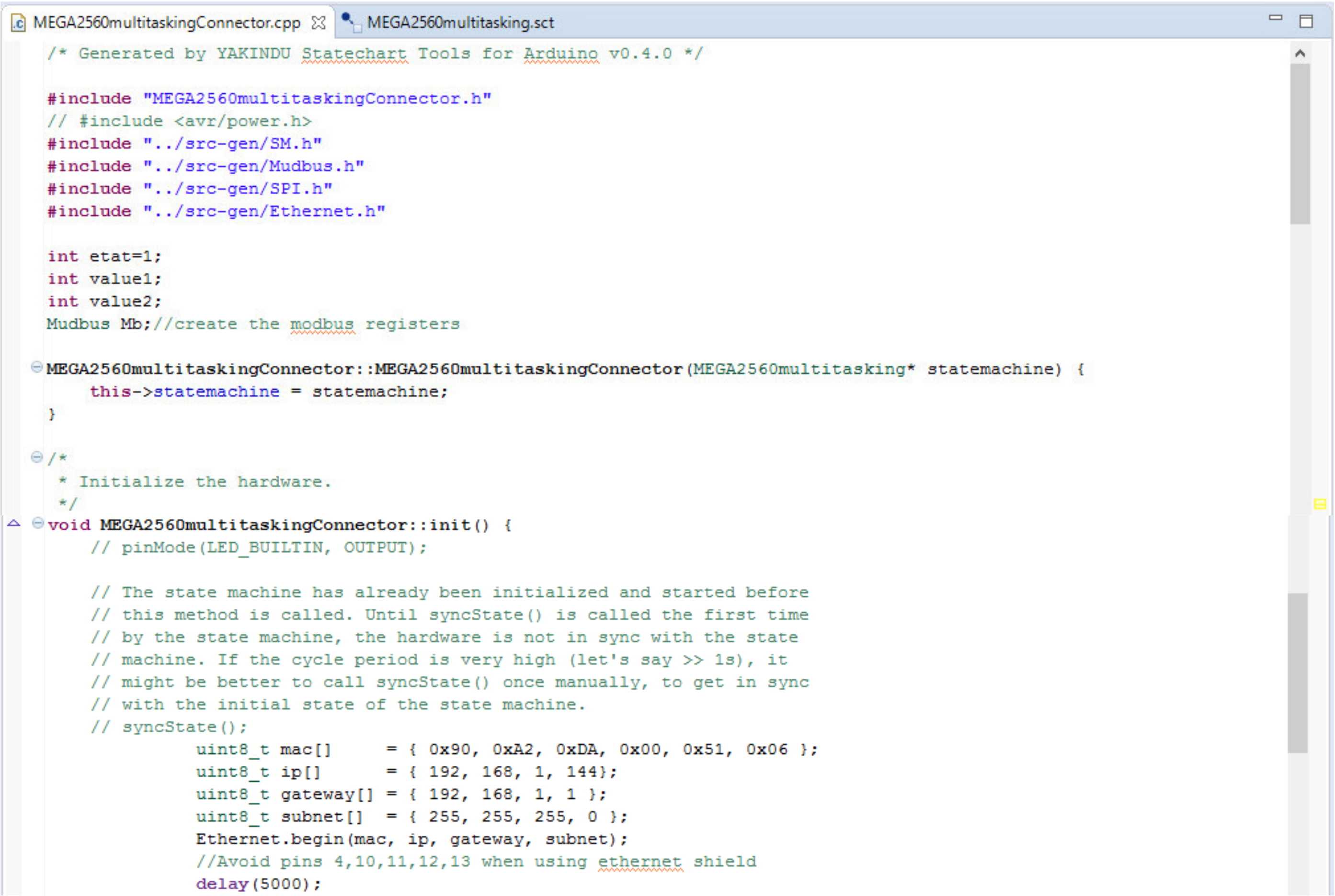


MEGA2560multitaskingConnector.cpp
MEGA2560multitasking.sct
Statechart MEGA2560multitasking

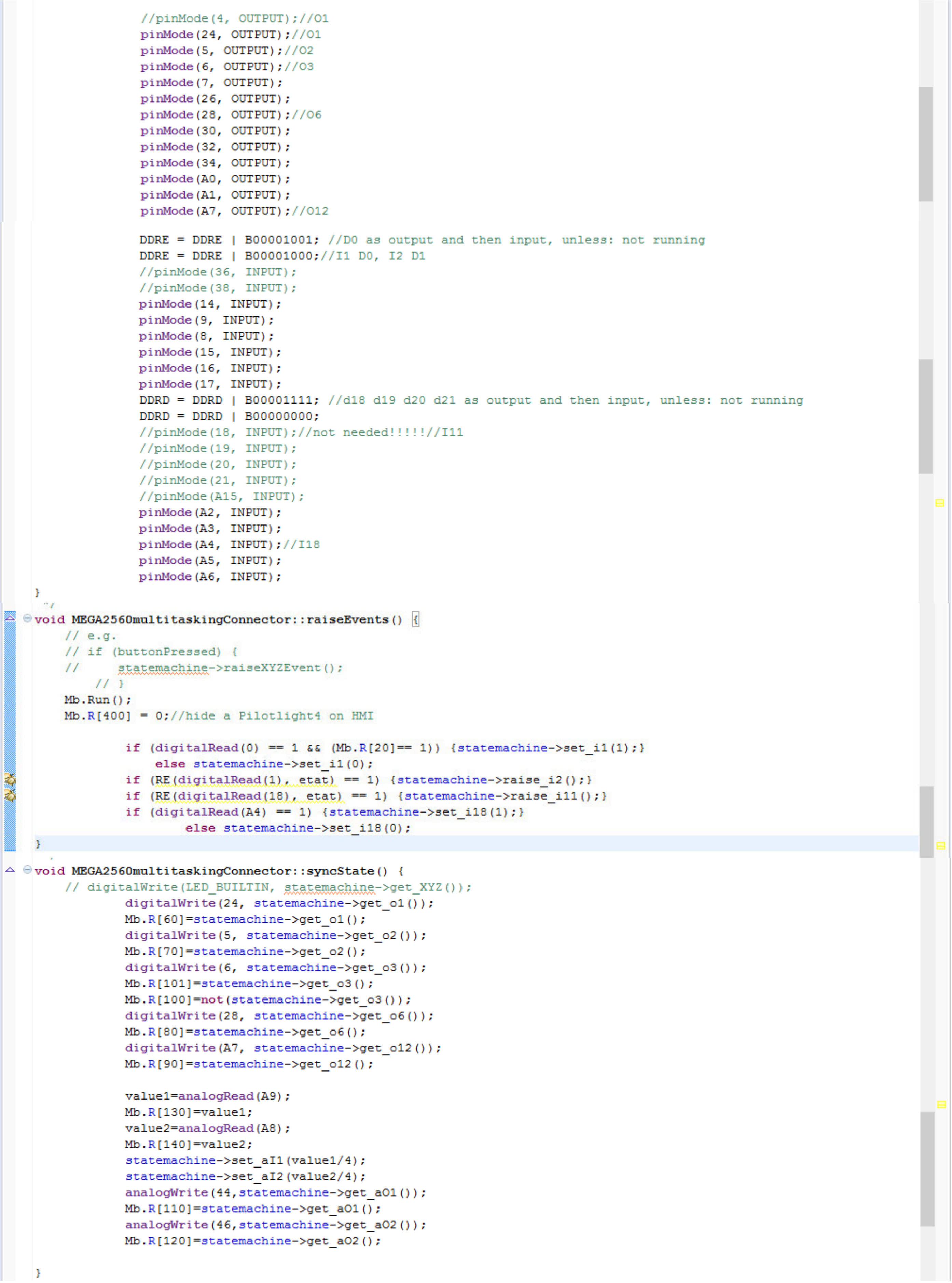
```
MEGA2560multitasking
interface:
// Define events and
// and variables here
var I1:integer
var I18:integer
in event I2
in event I11
var O1:boolean
var O2:boolean
var O3:boolean
var O6:boolean
var O12:integer
var AO1:integer
var AO2:integer
var AI1:integer
var AI2:integer
```

5-3 Modify and complete the generated code:

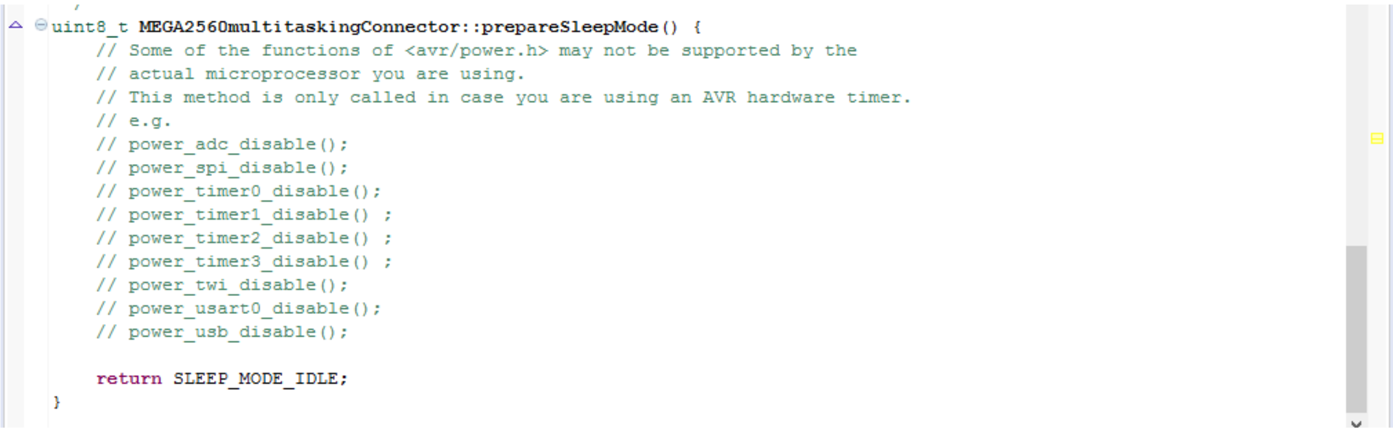
Use the switch to disconnect D4 (used by the Ethernet shield) and connect D24 instead (O1)



```
/* Generated by YAKINDU Statechart Tools for Arduino v0.4.0 */

#include "MEGA2560multitaskingConnector.h"
// #include <avr/power.h>
#include "../src-gen/SM.h"
#include "../src-gen/Mudbus.h"
#include "../src-gen/SPI.h"
#include "../src-gen/Ethernet.h"

int etat=1;
int value1;
int value2;
Mudbus Mb;//create the modbus registers

MEGA2560multitaskingConnector::MEGA2560multitaskingConnector(MEGA2560multitasking* statemachine) {
    this->statemachine = statemachine;
}

/*
 * Initialize the hardware.
 */
void MEGA2560multitaskingConnector::init() {
    // pinMode(LED_BUILTIN, OUTPUT);

    // The state machine has already been initialized and started before
    // this method is called. Until syncState() is called the first time
    // by the state machine, the hardware is not in sync with the state
    // machine. If the cycle period is very high (let's say >> 1s), it
    // might be better to call syncState() once manually, to get in sync
    // with the initial state of the state machine.
    // syncState();
            uint8_t mac[]     = { 0x90, 0xA2, 0xDA, 0x00, 0x51, 0x06 };
            uint8_t ip[]      = { 192, 168, 1, 144};
            uint8_t gateway[] = { 192, 168, 1, 1 };
            uint8_t subnet[]  = { 255, 255, 255, 0 };
            Ethernet.begin(mac, ip, gateway, subnet);
            //Avoid pins 4,10,11,12,13 when using ethernet shield
            delay(5000);
```

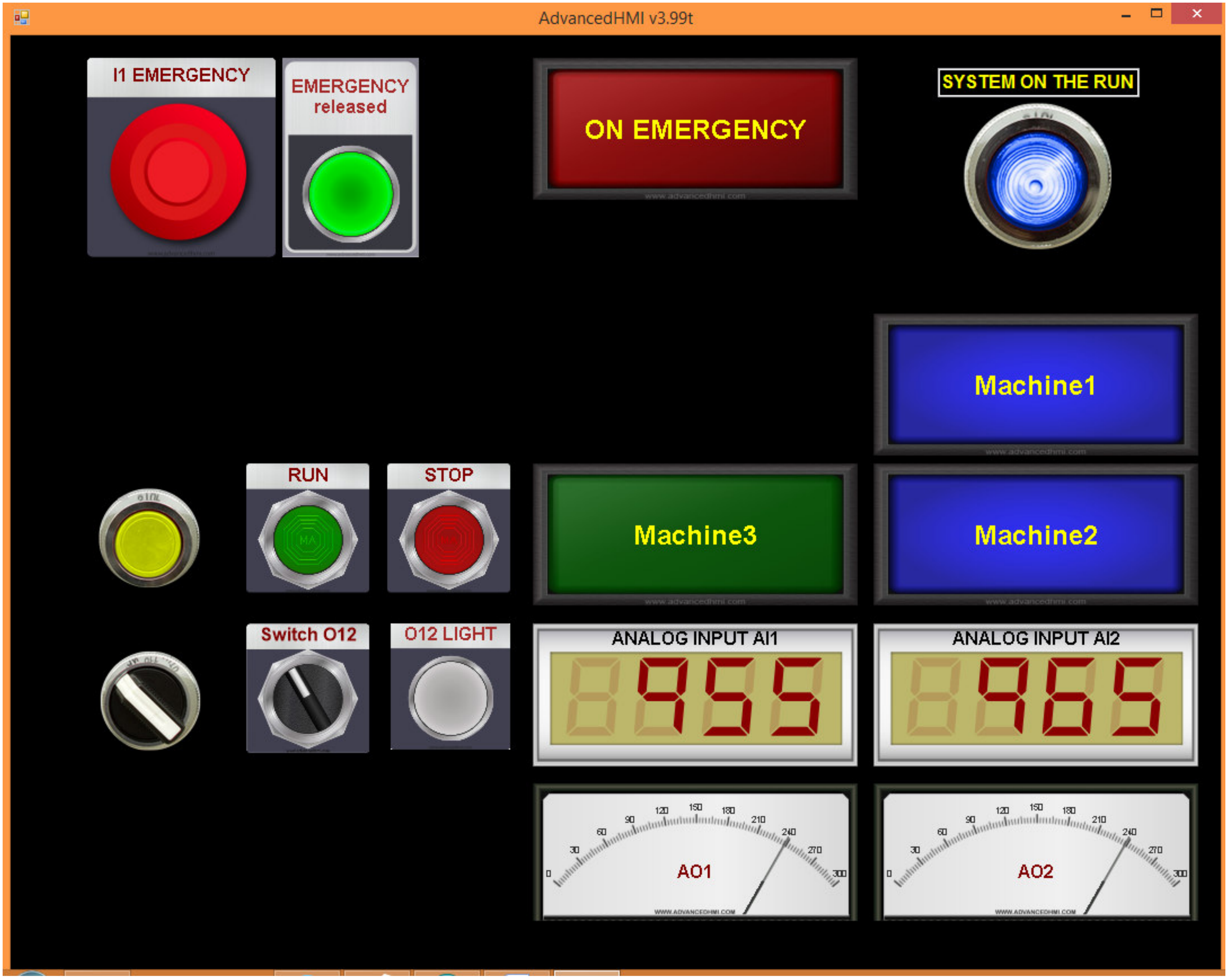


```
              //pinMode(4, OUTPUT);//O1
              pinMode(24, OUTPUT);//O1
              pinMode(5, OUTPUT);//O2
              pinMode(6, OUTPUT);//O3
              pinMode(7, OUTPUT);
              pinMode(26, OUTPUT);
              pinMode(28, OUTPUT);//O6
              pinMode(30, OUTPUT);
              pinMode(32, OUTPUT);
              pinMode(34, OUTPUT);
              pinMode(A0, OUTPUT);
              pinMode(A1, OUTPUT);
              pinMode(A7, OUTPUT);//O12

              DDRE = DDRE | B00001001; //D0 as output and then input, unless: not running
              DDRE = DDRE | B00001000;//I1 D0, I2 D1
              //pinMode(36, INPUT);
              //pinMode(38, INPUT);
              pinMode(14, INPUT);
              pinMode(9, INPUT);
              pinMode(8, INPUT);
              pinMode(15, INPUT);
              pinMode(16, INPUT);
              pinMode(17, INPUT);
              DDRD = DDRD | B00001111; //d18 d19 d20 d21 as output and then input, unless: not running
              DDRD = DDRD | B00000000;
              //pinMode(18, INPUT);//not needed!!!!!//I11
              //pinMode(19, INPUT);
              //pinMode(20, INPUT);
              //pinMode(21, INPUT);
              //pinMode(A15, INPUT);
              pinMode(A2, INPUT);
              pinMode(A3, INPUT);
              pinMode(A4, INPUT);//I18
              pinMode(A5, INPUT);
              pinMode(A6, INPUT);
}
 */
void MEGA2560multitaskingConnector::raiseEvents() {
    // e.g.
    // if (buttonPressed) {
    //     statemachine->raiseXYZEvent();
        // }
    Mb.Run();
    Mb.R[400] = 0;//hide a Pilotlight4 on HMI

            if (digitalRead(0) == 1 && (Mb.R[20]== 1)) {statemachine->set_i1(1);}
                else statemachine->set_i1(0);
            if (RE(digitalRead(1), etat) == 1) {statemachine->raise_i2();}
            if (RE(digitalRead(18), etat) == 1) {statemachine->raise_i11();}
            if (digitalRead(A4) == 1) {statemachine->set_i18(1);}
                    else statemachine->set_i18(0);
}

void MEGA2560multitaskingConnector::syncState() {
    // digitalWrite(LED_BUILTIN, statemachine->get_XYZ());
            digitalWrite(24, statemachine->get_o1());
            Mb.R[60]=statemachine->get_o1();
            digitalWrite(5, statemachine->get_o2());
            Mb.R[70]=statemachine->get_o2();
            digitalWrite(6, statemachine->get_o3());
            Mb.R[101]=statemachine->get_o3();
            Mb.R[100]=not(statemachine->get_o3());
            digitalWrite(28, statemachine->get_o6());
            Mb.R[80]=statemachine->get_o6();
            digitalWrite(A7, statemachine->get_o12());
            Mb.R[90]=statemachine->get_o12();

            value1=analogRead(A9);
            Mb.R[130]=value1;
            value2=analogRead(A8);
            Mb.R[140]=value2;
            statemachine->set_aI1(value1/4);
            statemachine->set_aI2(value2/4);
            analogWrite(44,statemachine->get_aO1());
            Mb.R[110]=statemachine->get_aO1();
            analogWrite(46,statemachine->get_aO2());
            Mb.R[120]=statemachine->get_aO2();

}
```



```
uint8_t MEGA2560multitaskingConnector::prepareSleepMode() {
    // Some of the functions of <avr/power.h> may not be supported by the
    // actual microprocessor you are using.
    // This method is only called in case you are using an AVR hardware timer.
    // e.g.
    // power_adc_disable();
    // power_spi_disable();
    // power_timer0_disable();
    // power_timer1_disable() ;
    // power_timer2_disable() ;
    // power_timer3_disable() ;
    // power_twi_disable();
    // power_usart0_disable();
    // power_usb_disable();

    return SLEEP_MODE_IDLE;
}
```

Upload it and launch the previous HMI you created (an exe file in the archives I gave to you):



And that’s it.