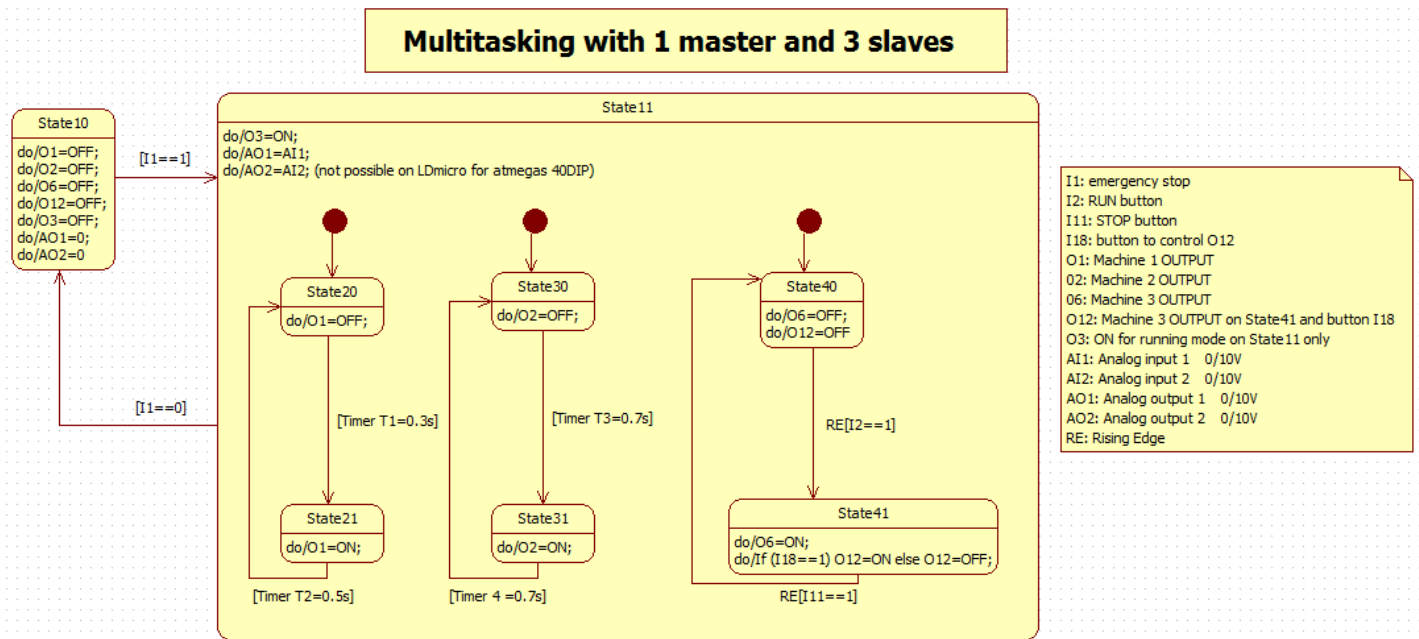


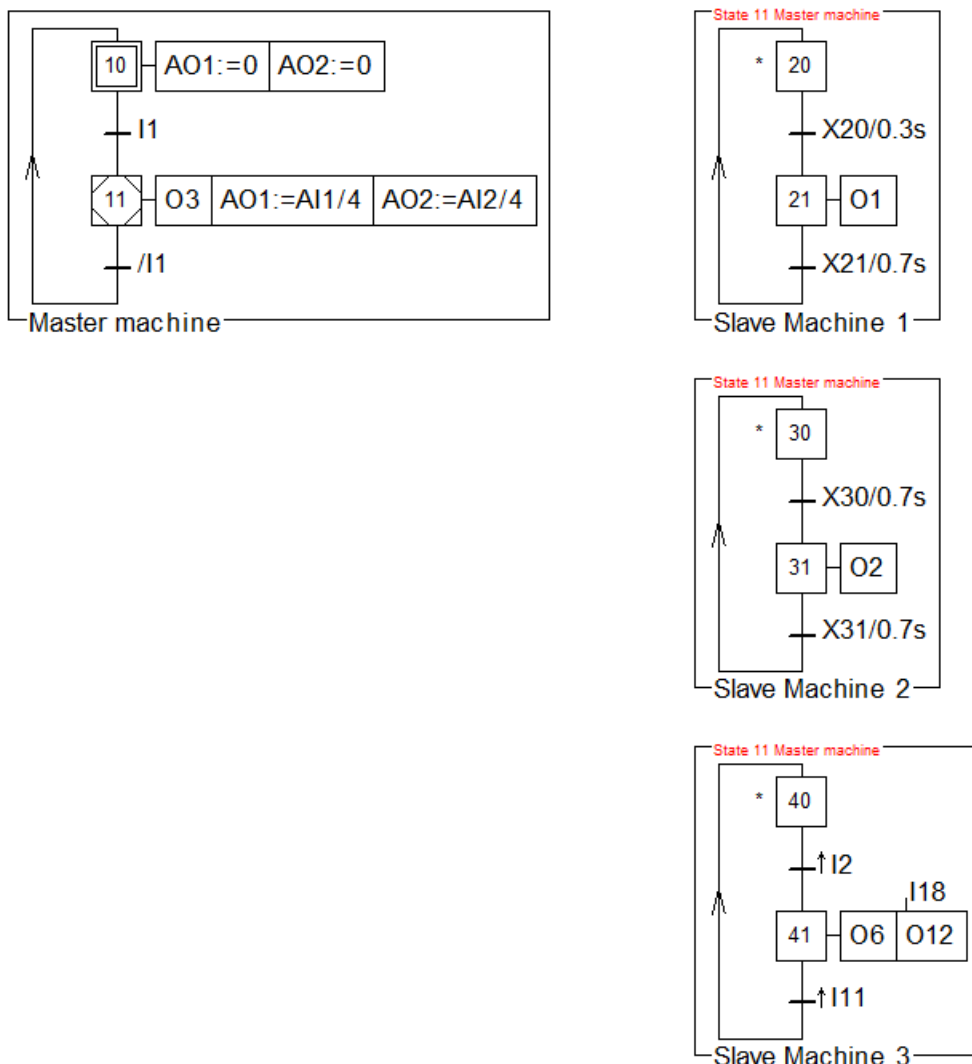
1-Multitasking with ladder programming on LDmicro4.2 for Atmega 40 pins DIP via USBasp

1-0 The machines to program:

The state diagram:



The SFC diagram: a bit different but the same multitasking, helps you to create the ladder



1-1 Settings on LDmicro:

PLC Configuration

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 program 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 control 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 clock before choosing a crystal.

LDmicro - Program Editor - C:\Users\Emmanuel\Desktop\arduino tom

File Edit Settings Instruction Simulate Compile Config Help

Microcontroller: Atmel AVR ATmega32 40-PDIP

Microcontrollers: TODO and DONE

```

0001 ;
7
0
0002 RState11 XI1
18 ] [ ] [
28
0003 RState11 XI1
18 ] [ ] [
28
0004 RB11
50 ] [ ] [
120
RState10 RState11
] [ ] [ ] [
RState10 RB10
] [ ] [ ] [
RB10
0005 ] [ ] [
36 RState11 RB11
90 ] [ ] [ ] [
207

```

Name	Type	State
XI1	digital in	0
XI11	digital in	0
XI18	digital in	0
XI2	digital in	0
YO1	digital out	0
YO12	digital out	0
YO2	digital out	0
YO3	digital out	0
YO6	digital out	0
RB10	int. relay	0
RB11	int. relay	0
RB20	int. relay	0
RB21	int. relay	0
RB30	int. relay	0
RB31	int. relay	0
RB40	int. relay	0
RB41	int. relay	0

modified Atmel AVR ATmega32 40-PDIP processor clock 16 MHz Tcycle=10 ms F=100 Hz F/

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

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

The master machine for emergency stop.

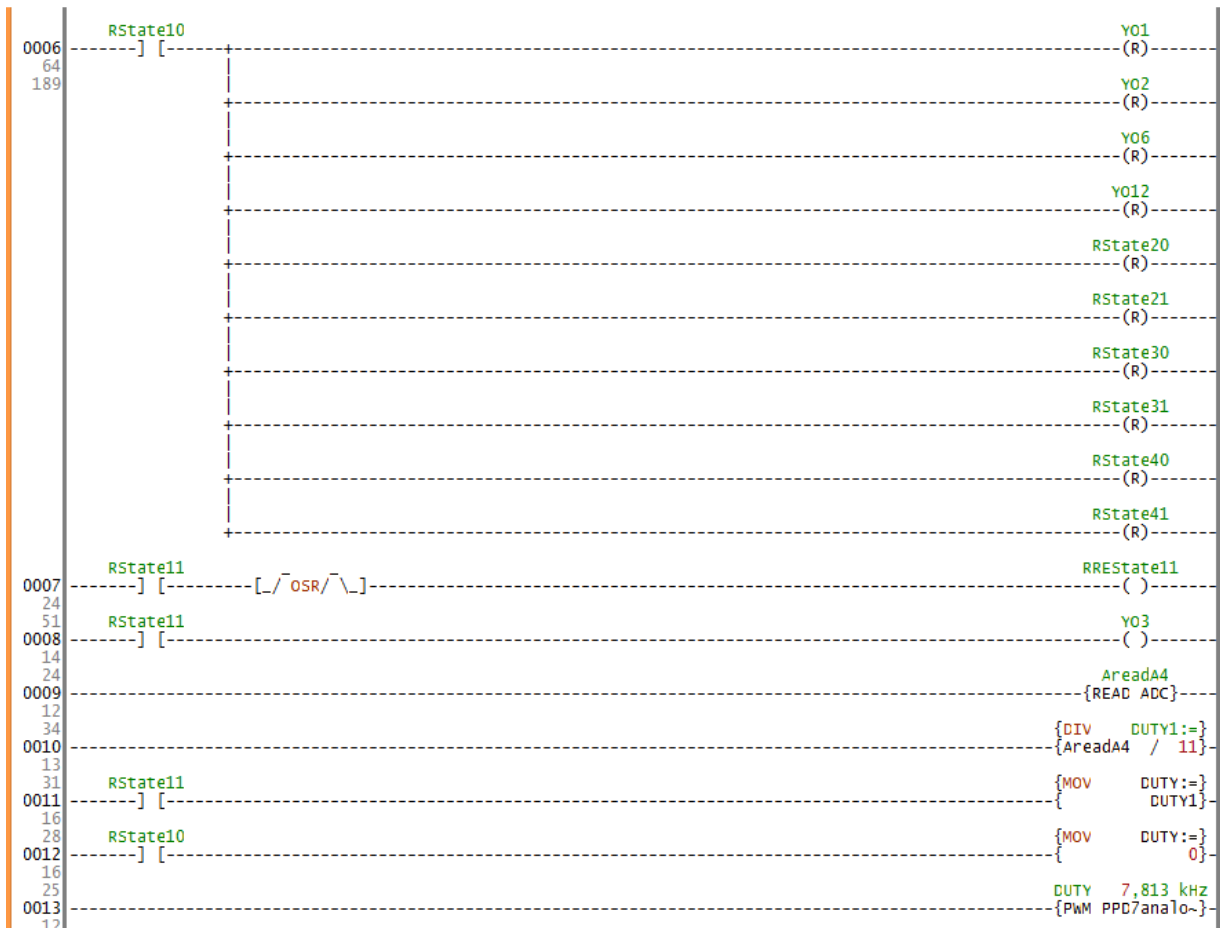
LDmicro - Program Editor - C:\Users\Emmanuel\Desktop\arduino tomaton\atmega32multitask.ld

File Edit Settings Instruction Simulate Compile Config Help

```

; Master machine
0001
7
0
0002 RState10 XI1 RB10
18 ] [ ] [ ] [
28
0003 RState11 XI1 RB11
18 ] [ ] [ ] [
28
0004 RB11 RState10
50 ] [ ] [ ] [
120
RState10 RState11
] [ ] [ ] [
RState10 RB10
] [ ] [ ] [
RB10
0005 ] [ ] [
36 RState11 RB11
90 ] [ ] [ ] [

```



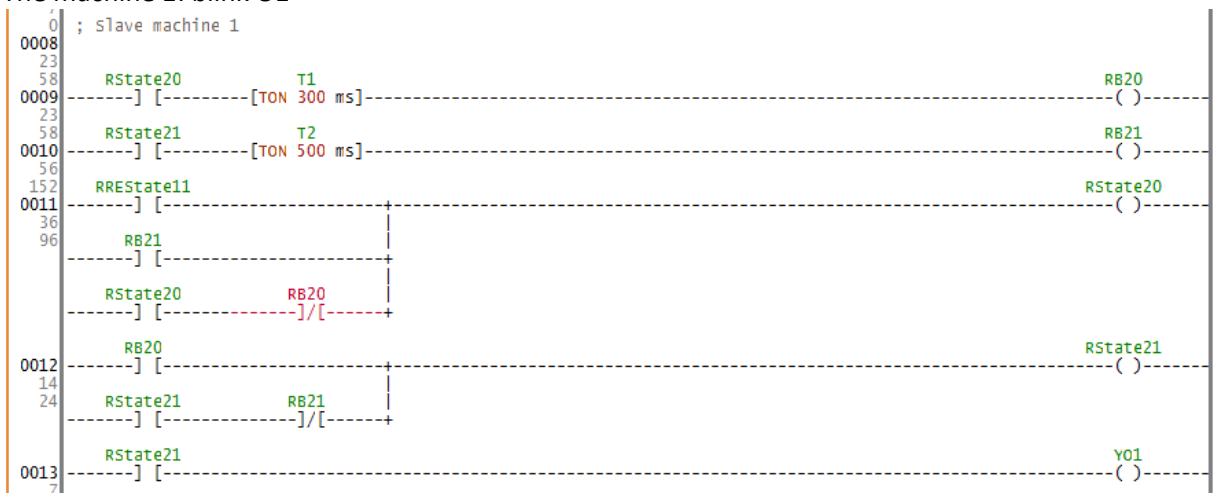
On State10 every outputs must be reset and all the states of the slaves machines too.

A rising edge on State11 event (OSR function) launches the 3 slaves.

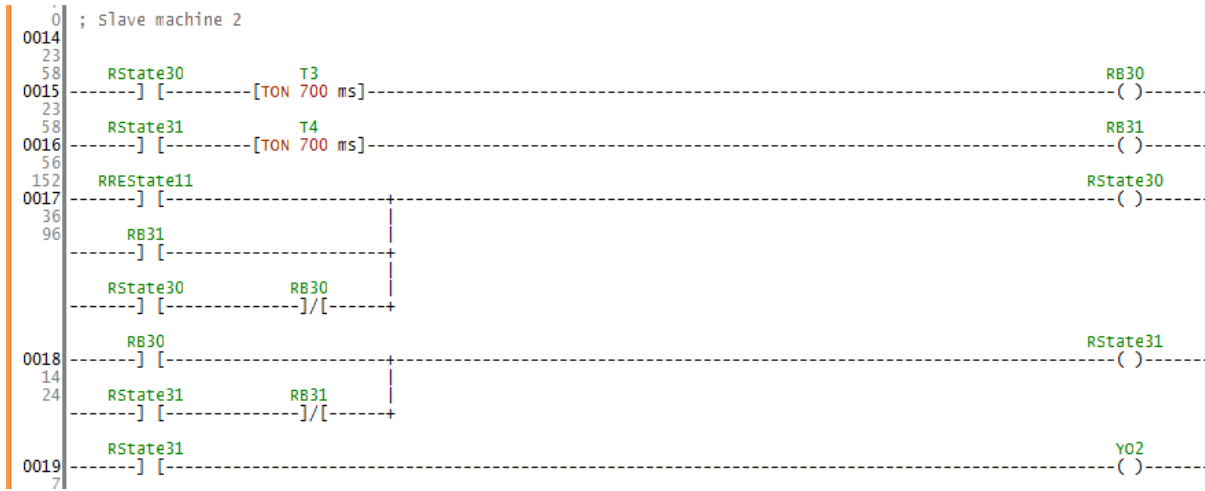
At the end of the script: AO1 receive AI1 value from the trimer. There is only one analog output available on LDmicro pin map of the atmegas 40dip. **Be careful! the variable used for readadc is "Axxxx": you must add "A" before your variable and AREF and AVCC are linked to VCC.**

There is only one analog output available on LDmicro on pin21 (PD7)

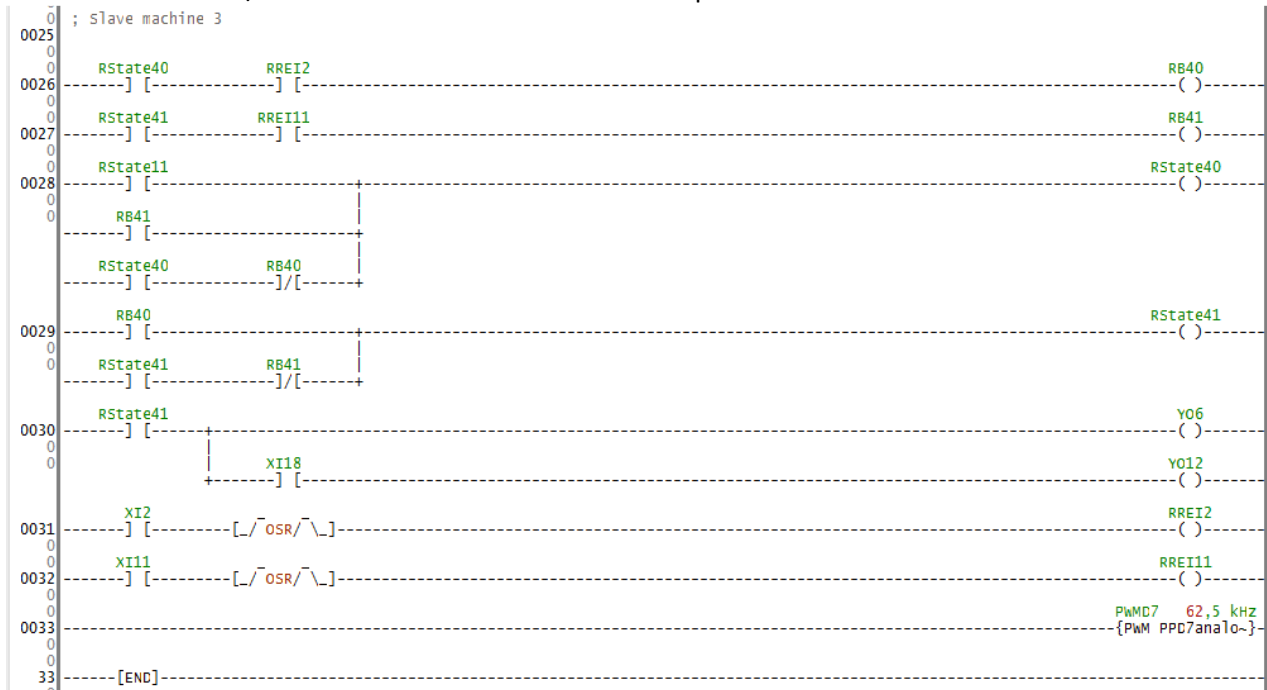
The machine 1: blink O1



The machine 2: blink O2



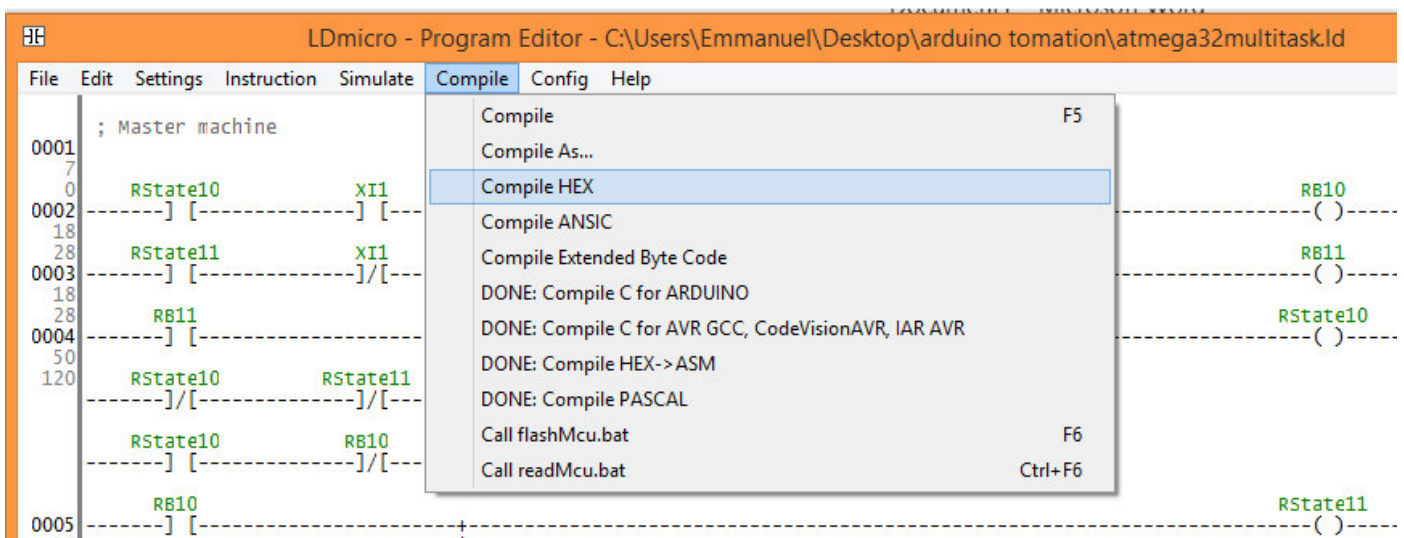
The Machine 3: RUN/STOP O6 and switch ON O12 if I18 is pushed within the State41.

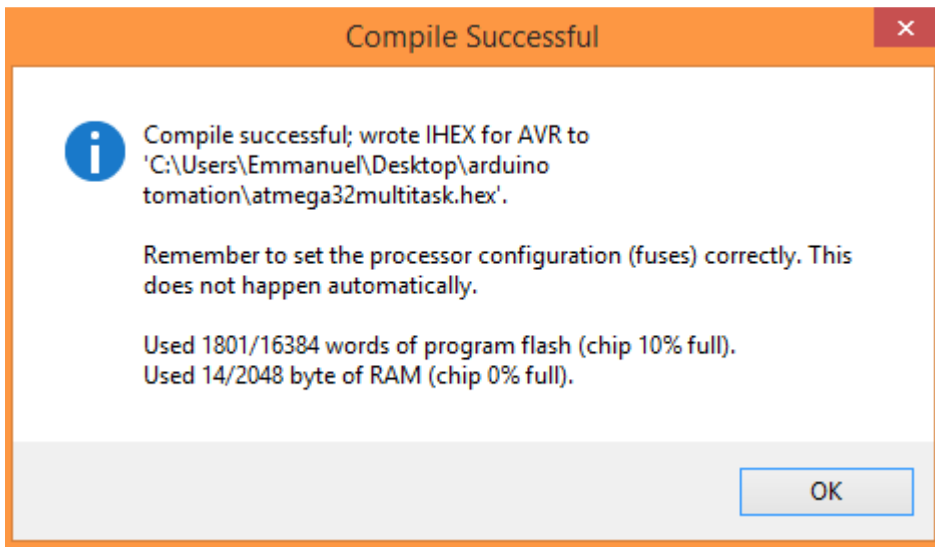


The list of INPUT/OUTPUT addresses on the microcontroller and the internal relays used in the ladder:

Name	Type	State	Pin on M...	MCU P...	Pin Name	Address	Size
DUTY	general var	0x0000 = 0				0x69	2 bytes
DUTY1	general var	0x0000 = 0				0x67	2 bytes
XI1	digital in	0	14	PD0		0x30 (BIT0)	1 bit
XI11	digital in	0	25	PC3		0x33 (BIT3)	1 bit
XI18	digital in	0	35	PA5		0x39 (BIT5)	1 bit
XI2	digital in	0	15	PD1		0x30 (BIT1)	1 bit
YO1	digital out	0	1	PB0		0x38 (BIT0)	1 bit
YO12	digital out	0	38	PA2		0x3b (BIT2)	1 bit
YO2	digital out	0	2	PB1		0x38 (BIT1)	1 bit
YO3	digital out	0	3	PB2		0x38 (BIT2)	1 bit
YO6	digital out	0	6	PB5		0x38 (BIT5)	1 bit
AreadA4	adc input	0x0000 = 0	36	PA4			1 pin/2...
PPD7analogOUT	PWM out	PWM	21	PD7		0x32 (BIT7)	1 pin
RB10	int. relay	0				0x60 (BIT2)	1 bit
RB11	int. relay	0				0x60 (BIT4)	1 bit
RB20	int. relay	0				0x62 (BIT3)	1 bit
RB21	int. relay	0				0x62 (BIT4)	1 bit
RB30	int. relay	0				0x6f (BIT1)	1 bit
RB31	int. relay	0				0x6f (BIT2)	1 bit
RB40	int. relay	0					1 bit
RB41	int. relay	0				0x74 (BIT2)	1 bit
RREI11	int. relay	0				0x74 (BIT1)	1 bit
RREI2	int. relay	0				0x6f (BIT7)	1 bit
RREState11	int. relay	0				0x62 (BIT1)	1 bit
RState10	int. relay	0				0x60 (BIT1)	1 bit
RState11	int. relay	0				0x60 (BIT3)	1 bit
RState20	int. relay	0				0x61 (BIT2)	1 bit
RState21	int. relay	0				0x61 (BIT3)	1 bit
RState30	int. relay	0				0x61 (BIT4)	1 bit
RState31	int. relay	0				0x61 (BIT5)	1 bit
RState40	int. relay	0				0x61 (BIT6)	1 bit
RState41	int. relay	0				0x61 (BIT7)	1 bit
T1	turn-on delay	0x0000 = 0 = 0 ms				0x6b	2 bytes
T2	turn-on delay	0x0000 = 0 = 0 ms				0x6d	2 bytes
T3	turn-on delay	0x0000 = 0 = 0 ms				0x70	2 bytes
T4	turn-on delay	0x0000 = 0 = 0 ms				0x72	2 bytes

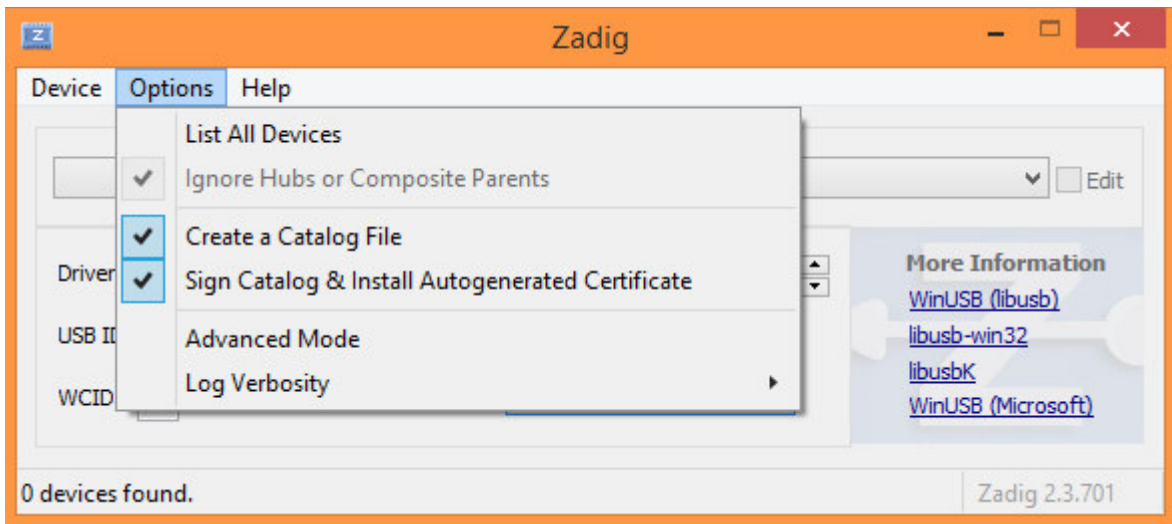
1-3 Compile the program in .hex file:



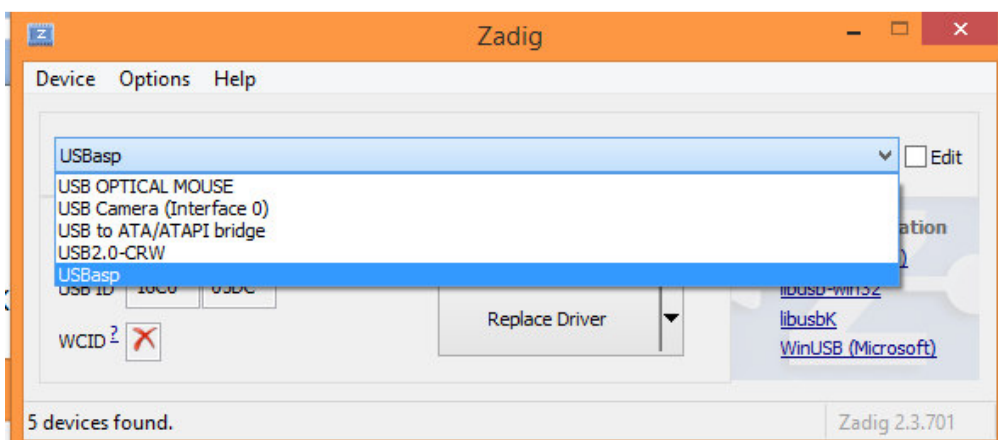


1-4 Launch Kazahama downloader:

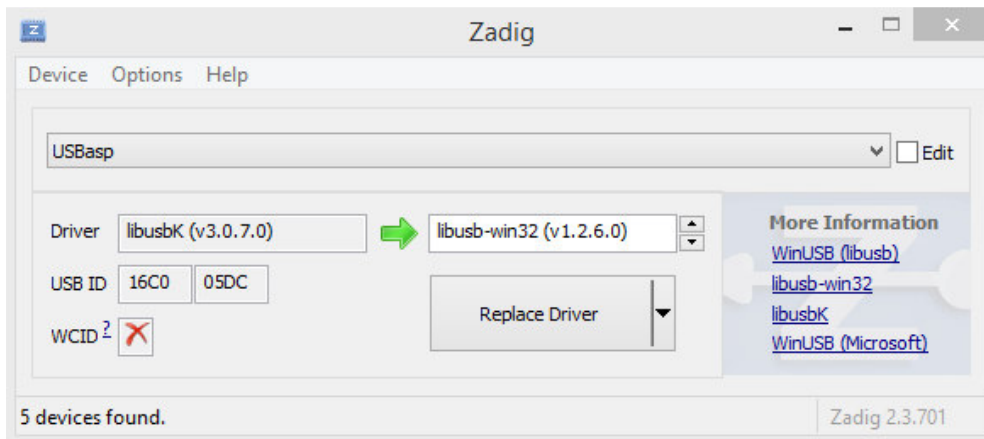
If you use arduino UNO board you need Xloader and for an arduino clone it's KAZHAMA. If KAZHAMA is not correctly launch due to a "Kernell32.dll error" or else, there is a problem of driver. So use Zadig program to correct the problem:



List All Devices



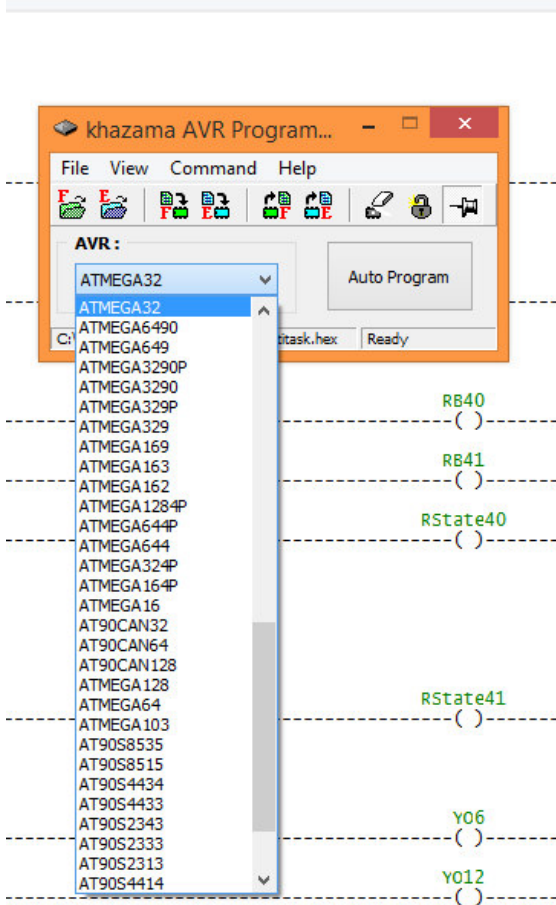
This one is the good one libusbk or libusb: replace or install driver



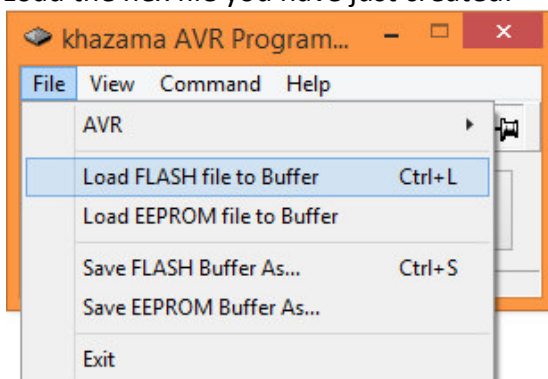
Now Khazama can run correctly.

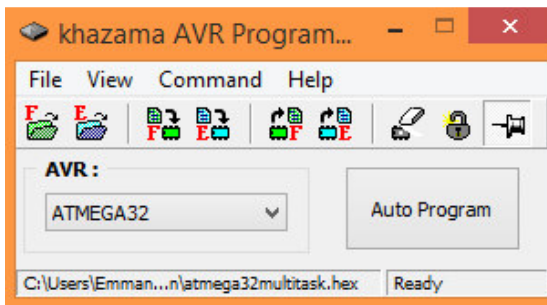
Select the good Microcontroller:

esktop\arduino\tomation\atmega32\multitask.id

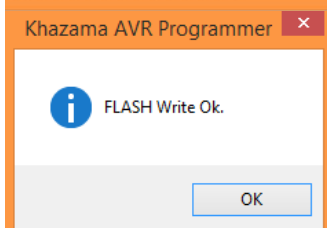
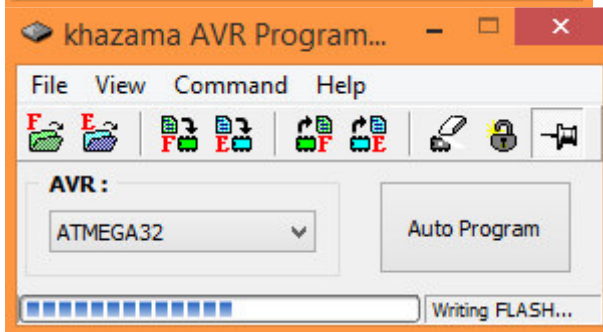
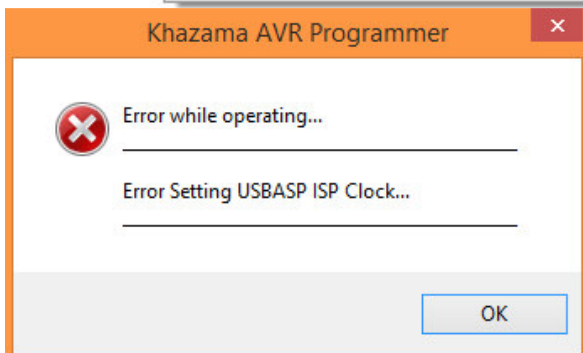
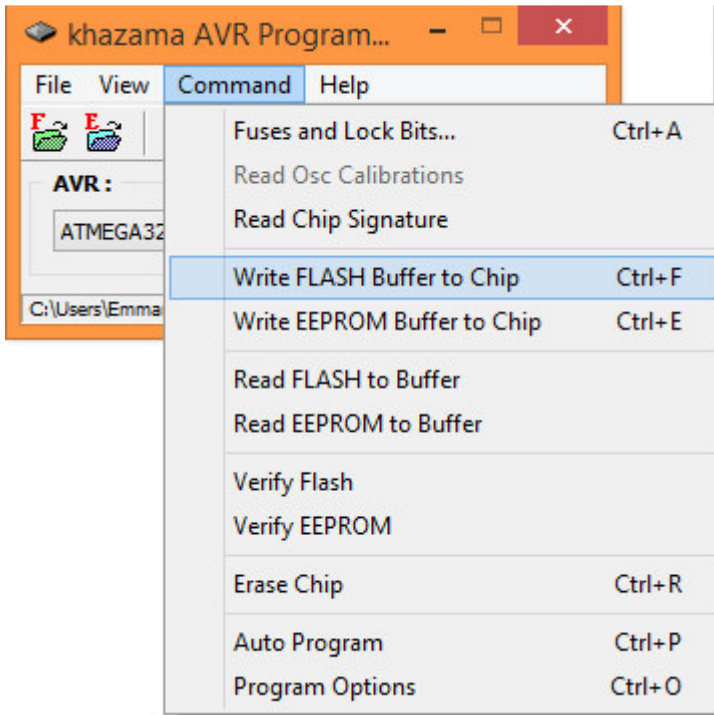


Load the hex file you have just created:





Switch off the 32I/O board on DOWNLOAD (to disconnect the SPI pins from the digital outputs).



Switch on the 32I/O board on DOWNLOAD (to reconnect the SPI pins to the digital outputs) and that's it and enjoy.

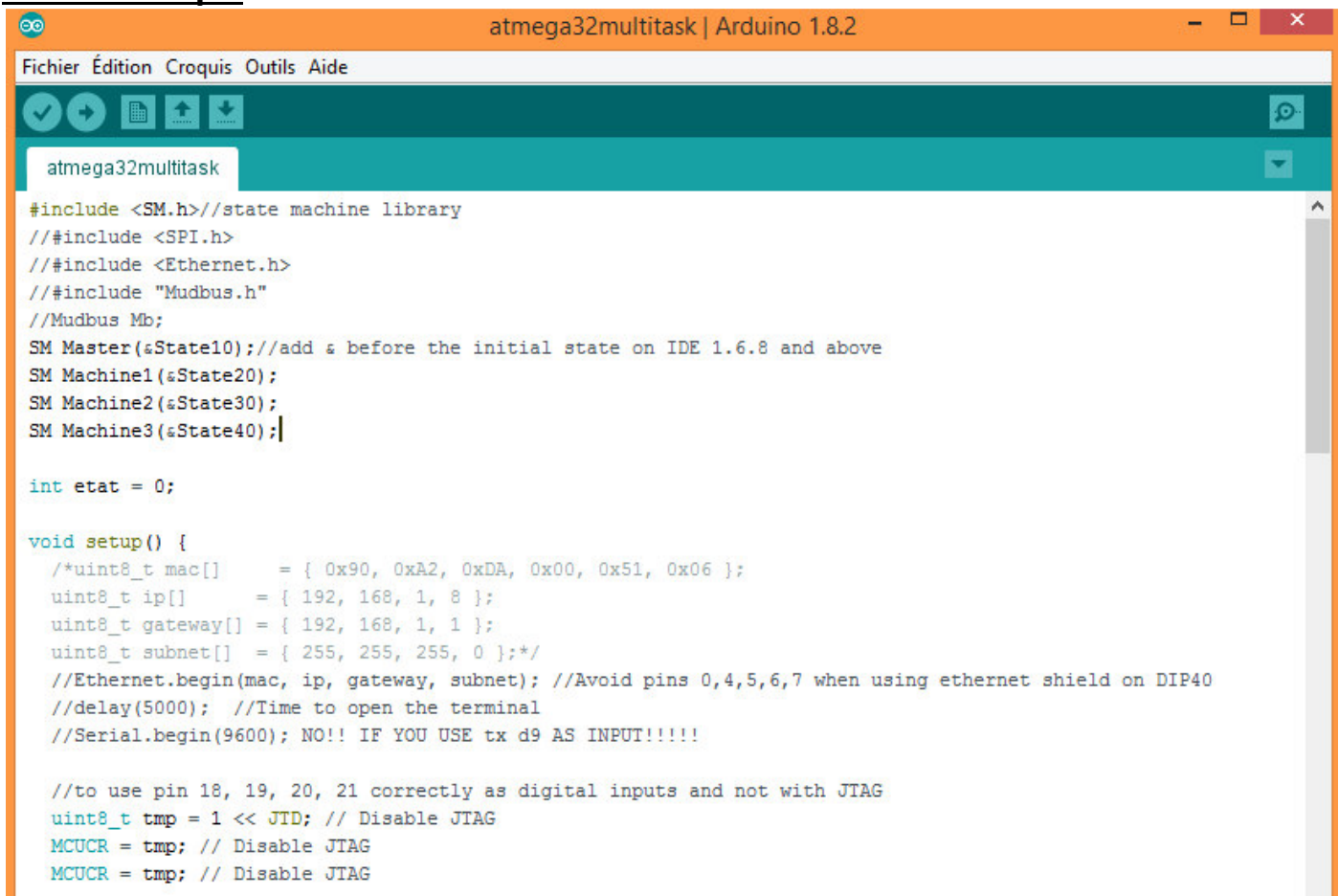
2-Multitasking with ladder programming on Arduino IDE 1.8.2 with SM library:

In order to use pins 18, 19, 20 and 21 on atmega 40 pin DIP you need to disable JTAG like this:

```
uint8_t tmp = 1 << JTD; // Disable JTAG
MCUCR = tmp; // Disable JTAG
MCUCR = tmp; // Disable JTAG
```

You need also to disable Serial and Ethernet whatever because all the I/O pins of the IC are used in particular SPI pins connected to digital outputs. So no Serial communication, no Ethernet shield with this board and no modbus TCP.

2-1 The script:



```
atmega32multitask

#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 pins 0,4,5,6,7 when using ethernet shield on DIP40
  //delay(5000); //Time to open the terminal
  //Serial.begin(9600); NO!! IF YOU USE tx d9 AS INPUT!!!!

  //to use pin 18, 19, 20, 21 correctly as digital inputs and not with JTAG
  uint8_t tmp = 1 << JTD; // Disable JTAG
  MCUCR = tmp; // Disable JTAG
  MCUCR = tmp; // Disable JTAG
```

In arduino 1.8.x you need to add "&" character before the first state of your machine. In this script pin 15 is used as analog output and pin A4 is used as analog input.

```

pinMode(1, OUTPUT);
pinMode(2, OUTPUT);
pinMode(3, OUTPUT);
pinMode(4, OUTPUT);
pinMode(5, OUTPUT);
pinMode(6, OUTPUT);
pinMode(7, OUTPUT);
pinMode(A0, OUTPUT);
pinMode(A1, OUTPUT);
//pinMode(15, OUTPUT);
pinMode(A2, OUTPUT);
DDRC = DDRC | B00000000;
pinMode(8, INPUT);
pinMode(9, INPUT);
pinMode(10, INPUT);
pinMode(11, INPUT);
pinMode(12, INPUT);
pinMode(13, INPUT);
pinMode(14, INPUT);
/*pinMode(16, INPUT);
  pinMode(17, INPUT);
  pinMode(18, INPUT);
  pinMode(19, INPUT);
  pinMode(20, INPUT);
  pinMode(21, INPUT);
  pinMode(22, INPUT);
  pinMode(23, INPUT);*/
pinMode(A7, INPUT);
pinMode(A6, INPUT);
pinMode(A5, INPUT);
//pinMode(A4, INPUT);
pinMode(A3, INPUT);
}

```

```

void loop() {
  EXEC(Master);
  if (digitalRead(8) == LOW) {
    Machine1.Finish(); Machine2.Finish(); Machine3.Finish();
  }
  if ((digitalRead(8) == 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 for emergency stop:

```
//The Master machine////////////////////////////////////
State State10() {
  digitalWrite(0, LOW);
  digitalWrite(1, LOW);
  digitalWrite(5, LOW);
  digitalWrite(A2, LOW);
  digitalWrite(2, LOW);
  analogWrite(15,0);
  if (digitalRead(8) == HIGH) Master.Set(State11);
}

State State11() {
  digitalWrite(2, HIGH);
  analogWrite(15, (analogRead(A4))/4);
  EXEC(Machine1);
  EXEC(Machine2);
  EXEC(Machine3);
  if (digitalRead(8) == LOW) Master.Set(State10);
}

```

The machine 1: blink O1

```
//The Machine1////////////////////////////////////
State State20() {
  digitalWrite(0, LOW);
  if (Machine1.Timeout(300)) Machine1.Set(State21) ;
}
State State21() {
  digitalWrite(0, HIGH);
  if (Machine1.Timeout(500)) Machine1.Set(State20) ;
}

```

The machine 2: blink O2

```
//The Machine2////////////////////////////////////
State State30() {
  digitalWrite(1, LOW);
  if (Machine2.Timeout(700)) Machine2.Set(State31) ;
}
State State31() {
  digitalWrite(1, HIGH);
  if (Machine2.Timeout(700)) Machine2.Set(State30) ;
}

```

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

```
//The Machine3////////////////////////////////////
State State40() {
  digitalWrite(5, LOW);
  digitalWrite(A2, LOW);
  if ((RE(digitalRead(9), etat) == 1)) Machine3.Set(State41) ;
}
State State41() {
  digitalWrite(5, HIGH);
  if ((digitalRead(A5) == HIGH))
    digitalWrite(A2, HIGH);
  else digitalWrite(A2, LOW);
  if ((RE(digitalRead(19), etat) == 1)) Machine3.Set(State40) ;
}

```

If there is no communication with USBasp, you need to launch Zadig in order to update the driver. You need arduino 1.8.x with mightycore inside.

2-2 Download the script:

1-Select the good microcontroller:

The screenshot shows the Arduino IDE interface for the 'atmega32multitask' project. The 'Type de carte' menu is open, listing various Arduino boards. The 'ATmega32' option is selected. The code editor in the background contains the following code:

```
#include <SM.h> // ...
// #include <SPI.h>
// #include <Ethernet.h>
// #include "Mudbus.h"
// Mudbus Mb;
SM Master(&State10);
SM Machine1(&State1);
SM Machine2(&State2);
SM Machine3(&State3);

int etat = 0;

void setup() {
  /*uint8_t mac[]
  uint8_t ip[]
  uint8_t gateway[]
  uint8_t subnet[]
  // Ethernet.begin(mac, ip, gateway, subnet); // Avoid pins 0, 4, 5, 6
  // delay(5000); // Time to open the terminal
  // Serial.begin(9600); NO!! IF YOU USE tx d9 AS INPUT!!!!

  // to use pin 18, 19, 20, 21 correctly as digital inputs and not
  uint8_t tmp = 1 << JTD; // Disable JTAG
  MCUCR = tmp; // Disable JTAG
  MCUCR = tmp; // Disable JTAG

  pinMode(0, OUTPUT);
  pinMode(1, OUTPUT);
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
```

The terminal at the bottom shows the following error messages:

```
Bibliothèque non valide trouvée dans C:\Users\Emmanuel\Documents\A
Bibliothèque non valide trouvée dans C:\Users\Emmanuel\Documents\A
Bibliothèque non valide trouvée dans C:\Users\Emmanuel\Documents\A
```

The status bar at the bottom indicates: "Mise à jour disponible pour certaines des vos cartes et bibliothèque".

2-Select the good programmer:

The screenshot shows the Arduino IDE interface for the 'atmega32multitask' project. The 'Tools' menu is open, and the 'Programmer' option is selected, which has opened a sub-menu. In this sub-menu, 'USBasp (MightyCore)' is highlighted with a mouse cursor. The main code editor shows the following code:

```
#include <SM.h> // ...
// #include <SPI.h>
// #include <Ethernet.h>
// #include "Mudbus.h"
// Mudbus Mb;
SM Master(&State10);
SM Machine1(&State2);
SM Machine2(&State3);
SM Machine3(&State4);

int etat = 0;

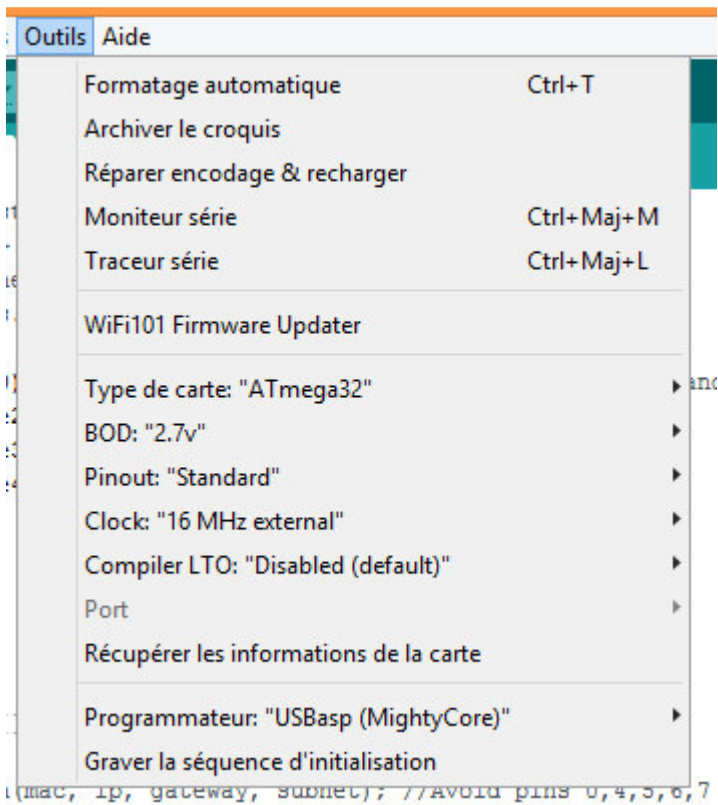
void setup() {
  /*uint8_t mac[]
  uint8_t ip[]
  uint8_t gateway[]
  uint8_t subnet[]
  //Ethernet.begin(mac, ip, gateway, subnet); //Avoid pins 0,4,5,6
  //delay(5000); //Time to open the terminal
  //Serial.begin(9600); NO!! IF YOU USE tx d9 AS INPUT!!!!

  //to use pin 18, 19, 20, 21 correctly as digital inputs and not
  uint8_t tmp = 1 << JTD; // Disable JTAG
  MCUCR = tmp; // Disable JTAG
  MCUCR = tmp; // Disable JTAG

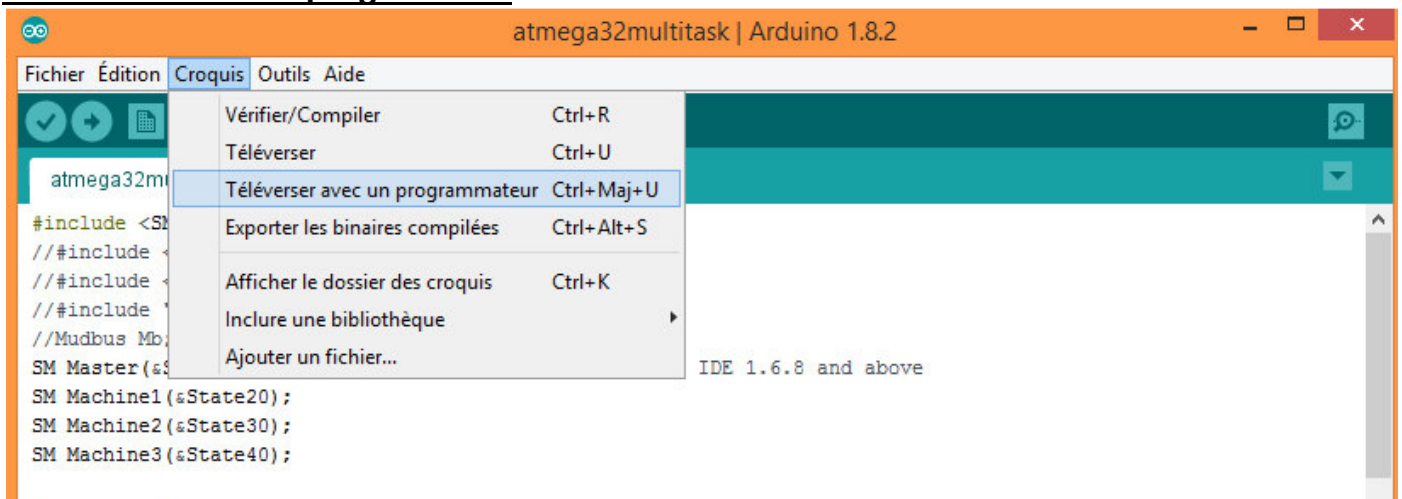
  pinMode(0, OUTPUT);
  pinMode(1, OUTPUT);
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
```

The 'Tools' menu options include: Formatage automatique (Ctrl+T), Archiver le croquis, Réparer encodage & recharger, Moniteur série (Ctrl+Maj+M), Traceur série (Ctrl+Maj+L), WiFi101 Firmware Updater, Type de carte: "ATmega32", BOD: "2.7v", Pinout: "Standard", Clock: "16 MHz external", Compiler LTO: "Disabled (default)", Port, Récupérer les informations de la carte, and Programmer: "USBasp (MightyCore)". The sub-menu for the programmer includes: AVR ISP, AVRISP mkII, USBtinyISP, ArduinoISP, ArduinoISP.org, USBasp, Parallel Programmer, Arduino as ISP, Arduino Gemma, BusPirate as ISP, Atmel STK500 development board, Atmel JTAGICE3 (ISP mode), Atmel JTAGICE3 (JTAG mode), Atmel-ICE (AVR), STK500 as ISP (MightyCore), AVR ISP (MightyCore), AVRISP mkII (MightyCore), USBtinyISP (MightyCore), ArduinoISP (MightyCore), USBasp (MightyCore), Parallel Programmer (MightyCore), Arduino as ISP (MightyCore), and BusPirate as ISP (MightyCore).

Done:



3-Download with a programmer:



That's it.