

Using Arduino with AUTOMGEN

(C)2014 IRAI

Release 5

Prerequisites

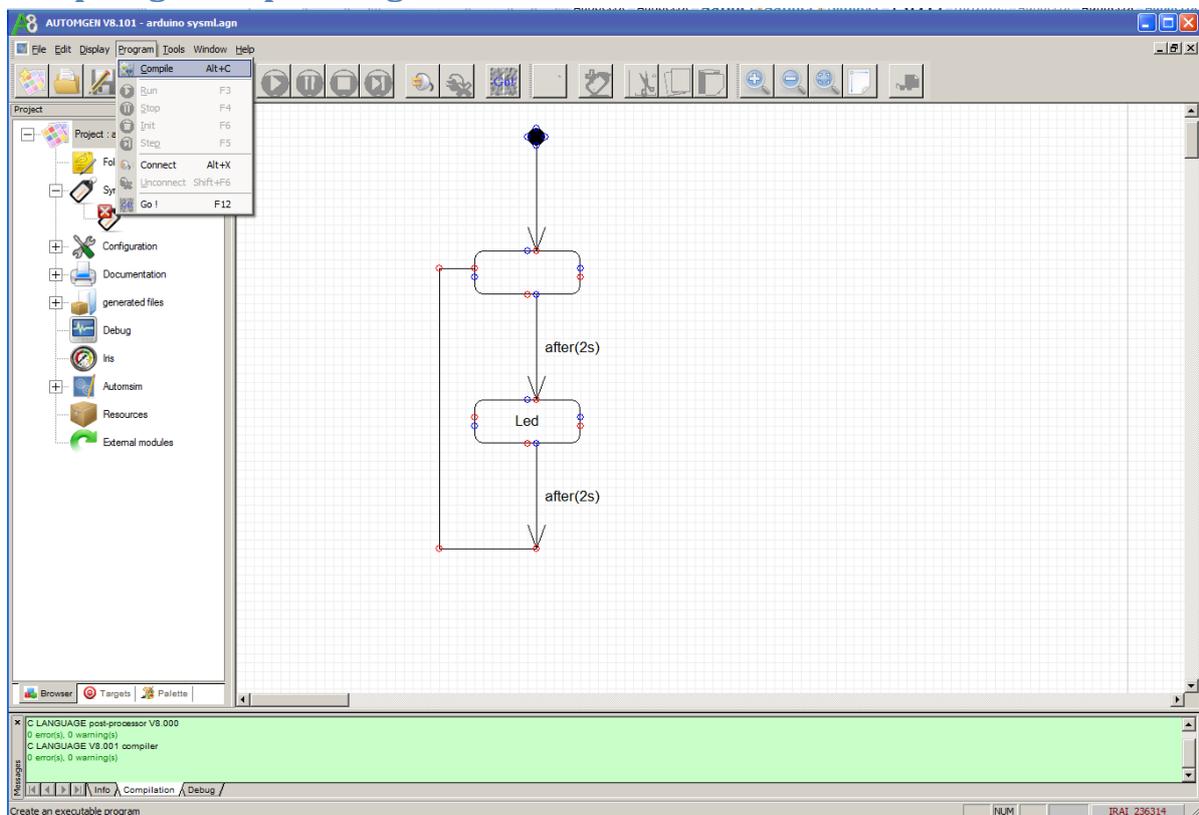
- Arduino 1.0 software
- AUTOMGEN 8.101

Main concept

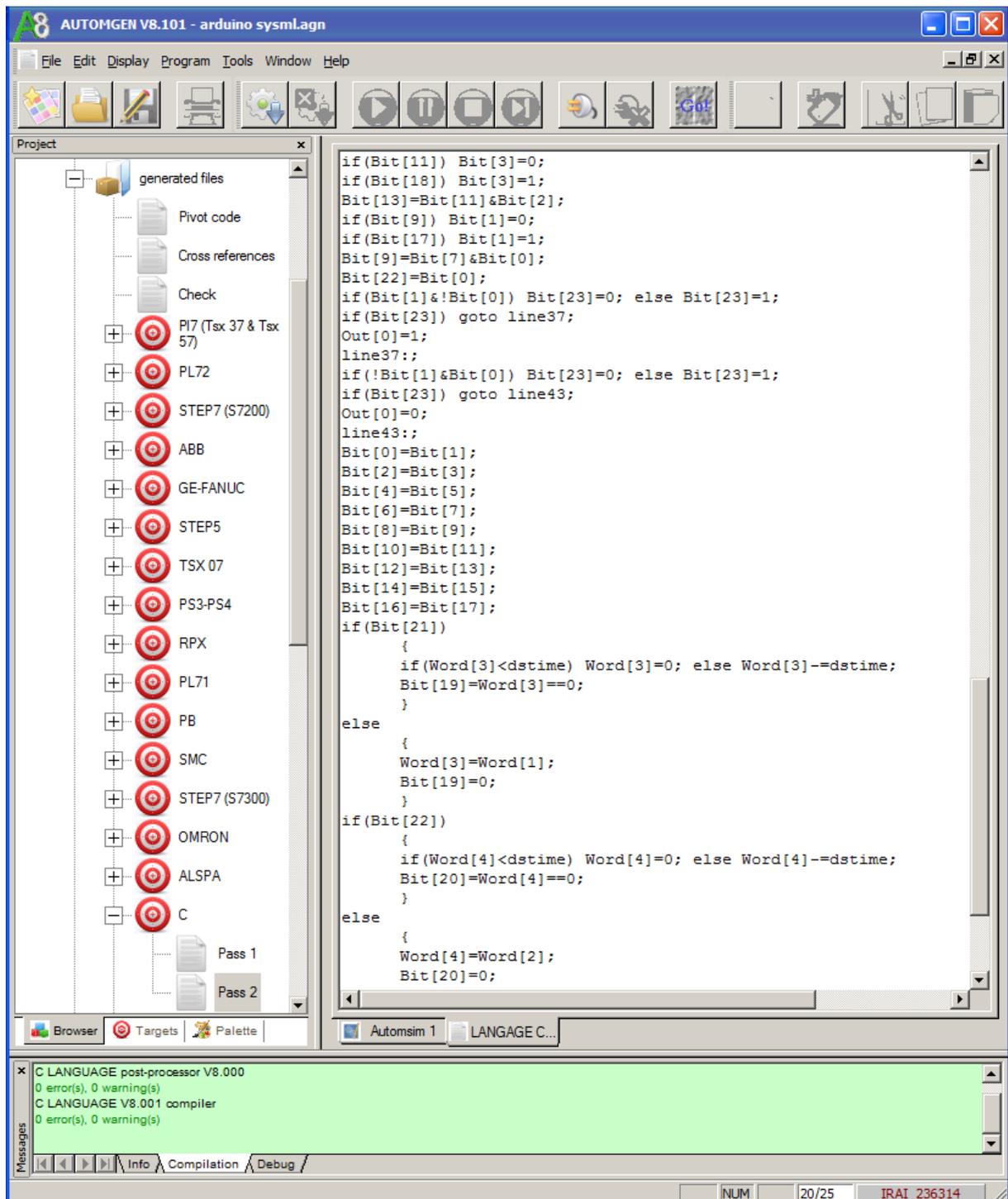
AUTOMGEN C language post-processor is used to generate code compatible with Arduino software. Two samples, one using Graftet, the other using SysML can be found in the "examples\post-processors\C" subdirectory of the AUTOMGEN installation directory. Two other samples demonstrate how to use connected mode and also automatic program upload feature.

For creating a new sample, please open one of the existing Arduino samples and modify it because all needed options are preset in them.

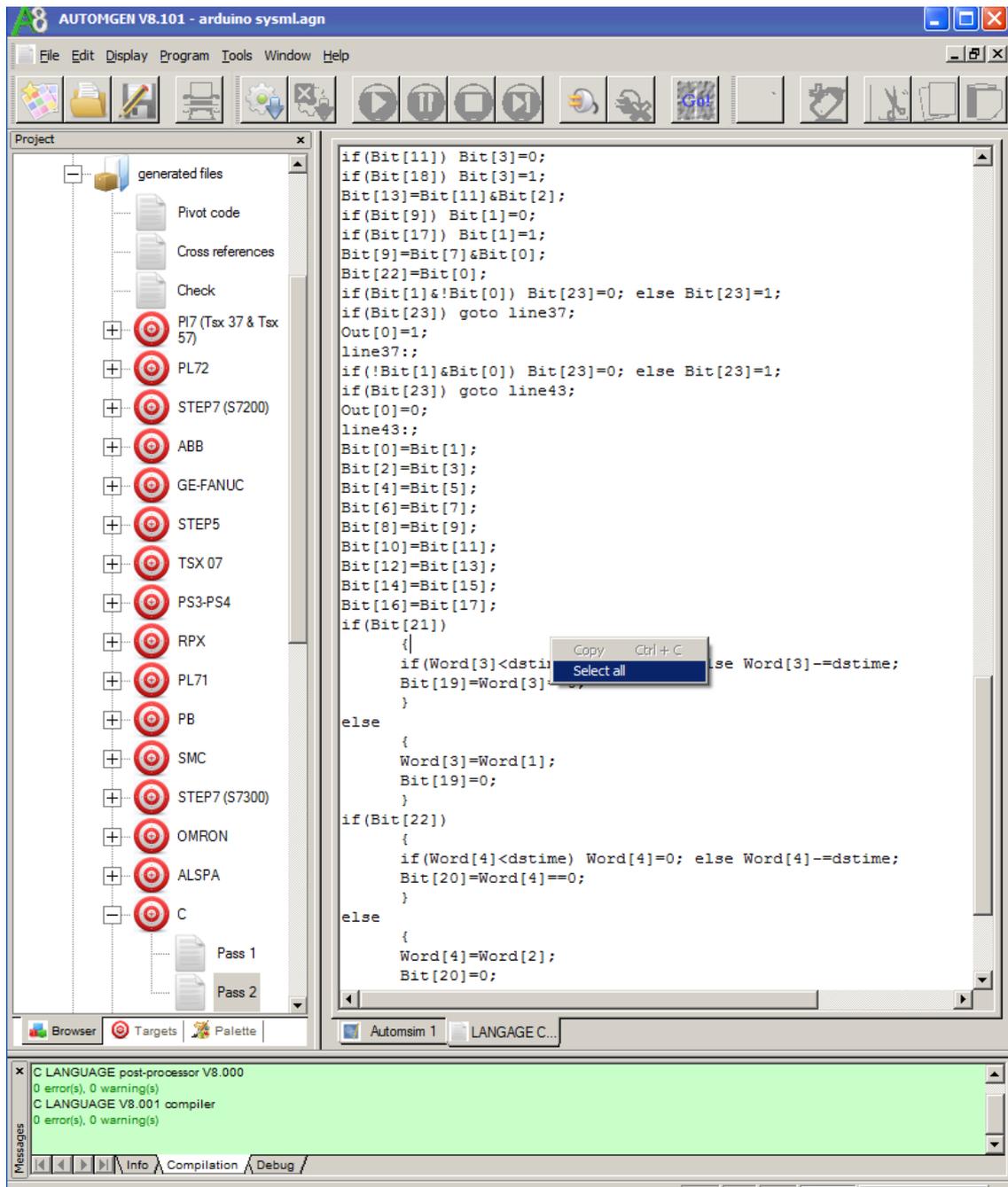
Compiling and uploading



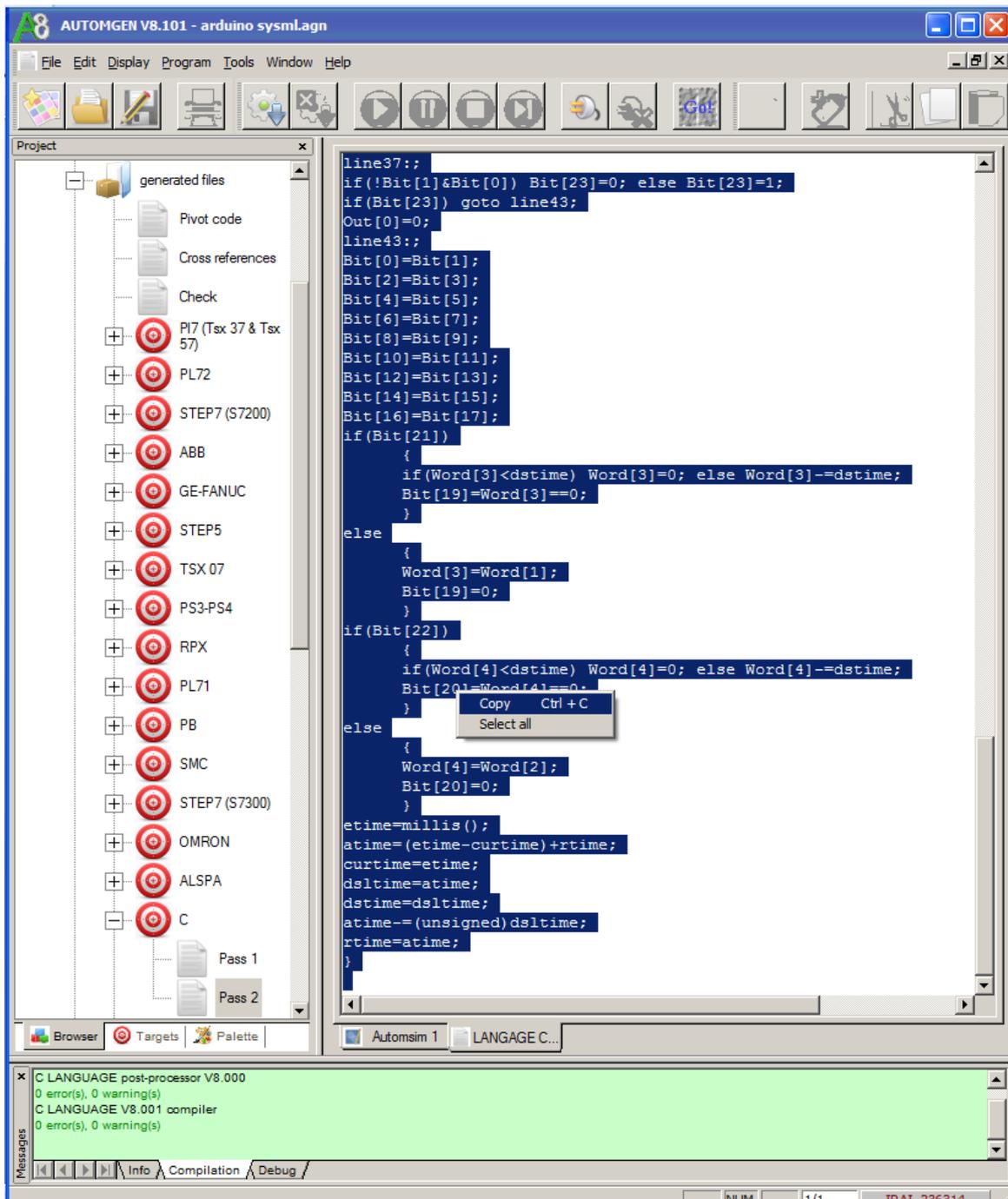
1- Compile program from AUTOMGEN



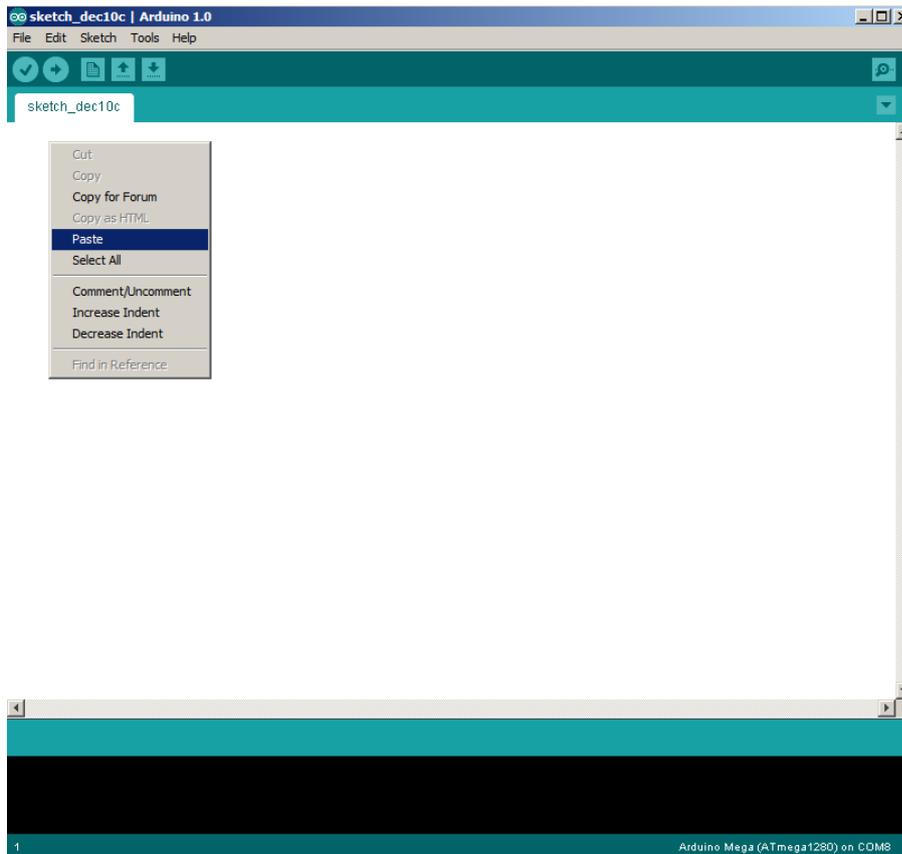
2- Open "generated files/C/Pass 2" item



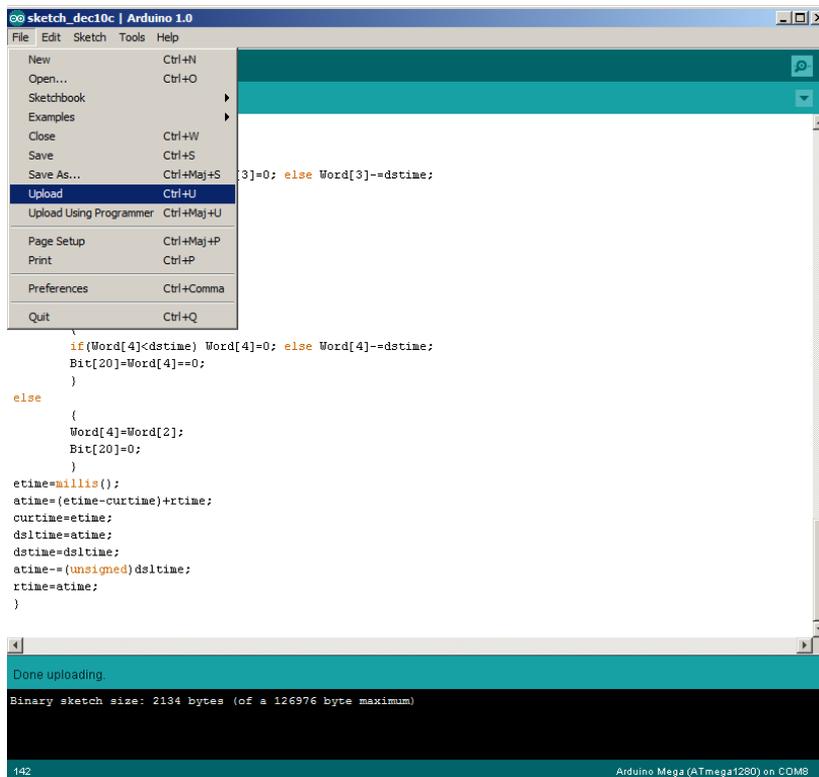
3- Select all generated code



4- copy the code



5- paste it in Arduino software



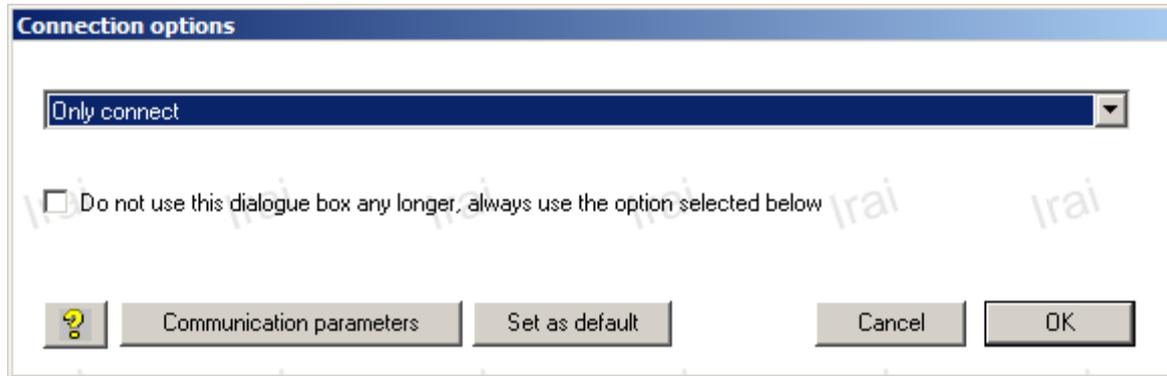
6- Upload the program, it will run automatically after the end of the upload

Remember to select the Arduino board on the Tool / Board menu in the Arduino software.

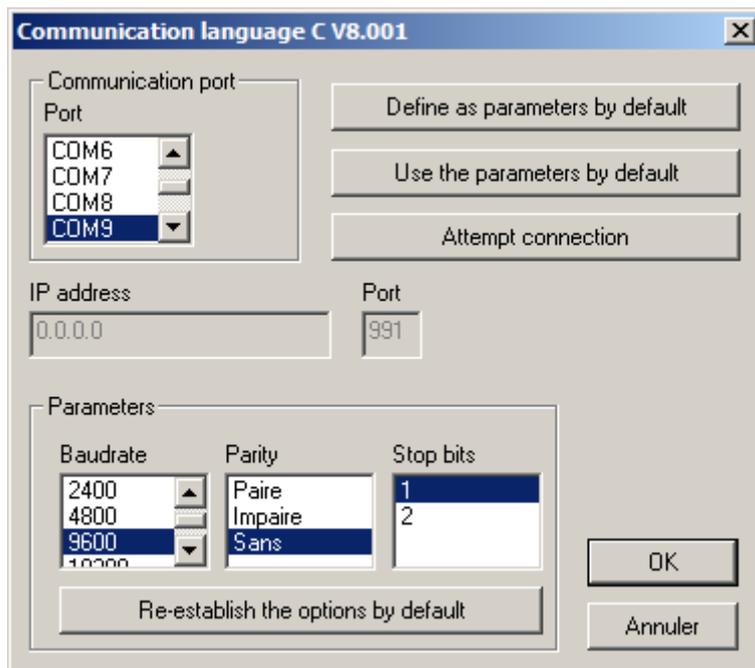
Connected mode

The "arduino debug.agn" sample present in "examples\Post-procesors\C" subdirectory of the AUTOMGEN installation directory shows how the connected mode (dynamic display, variables state modifications, step by step mode, etc.) is working.

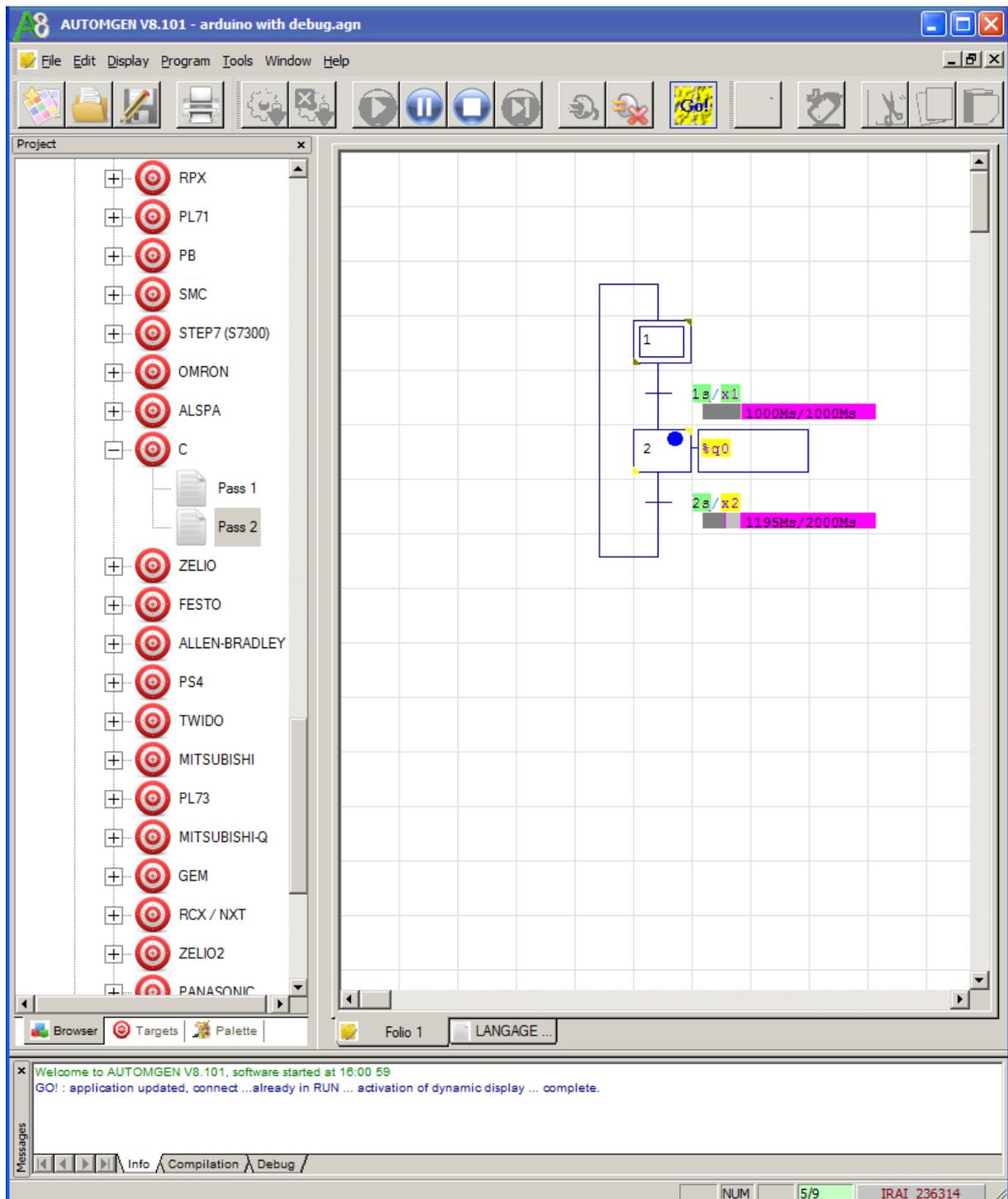
Communication parameters setup has to be made in the AUTOMGEN C post-processor dialog module properties dialog box:



"Only connect" connection mode have to be selected because upload process is managed by Arduino software.



Select COM port used to communicate with Arduino.

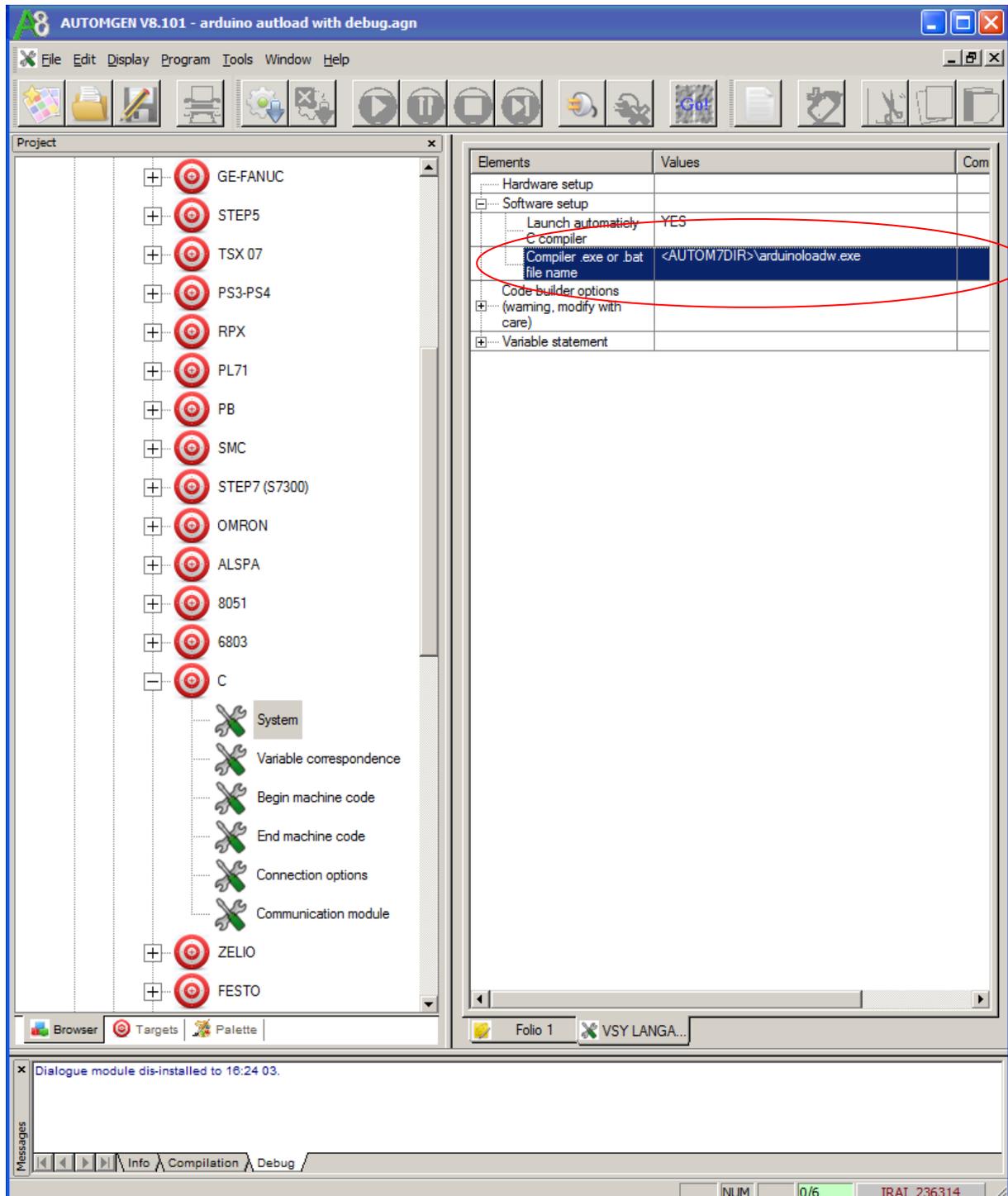


Automating the launch of the Arduino software

" arduino autload with debug.agn" " sample present in "examples\Post-processors\C" subdirectory of the AUTOMGEN installation directory shows, in addition to online mode, how to upload automatically code generated by AUTOMGEN into Arduino.

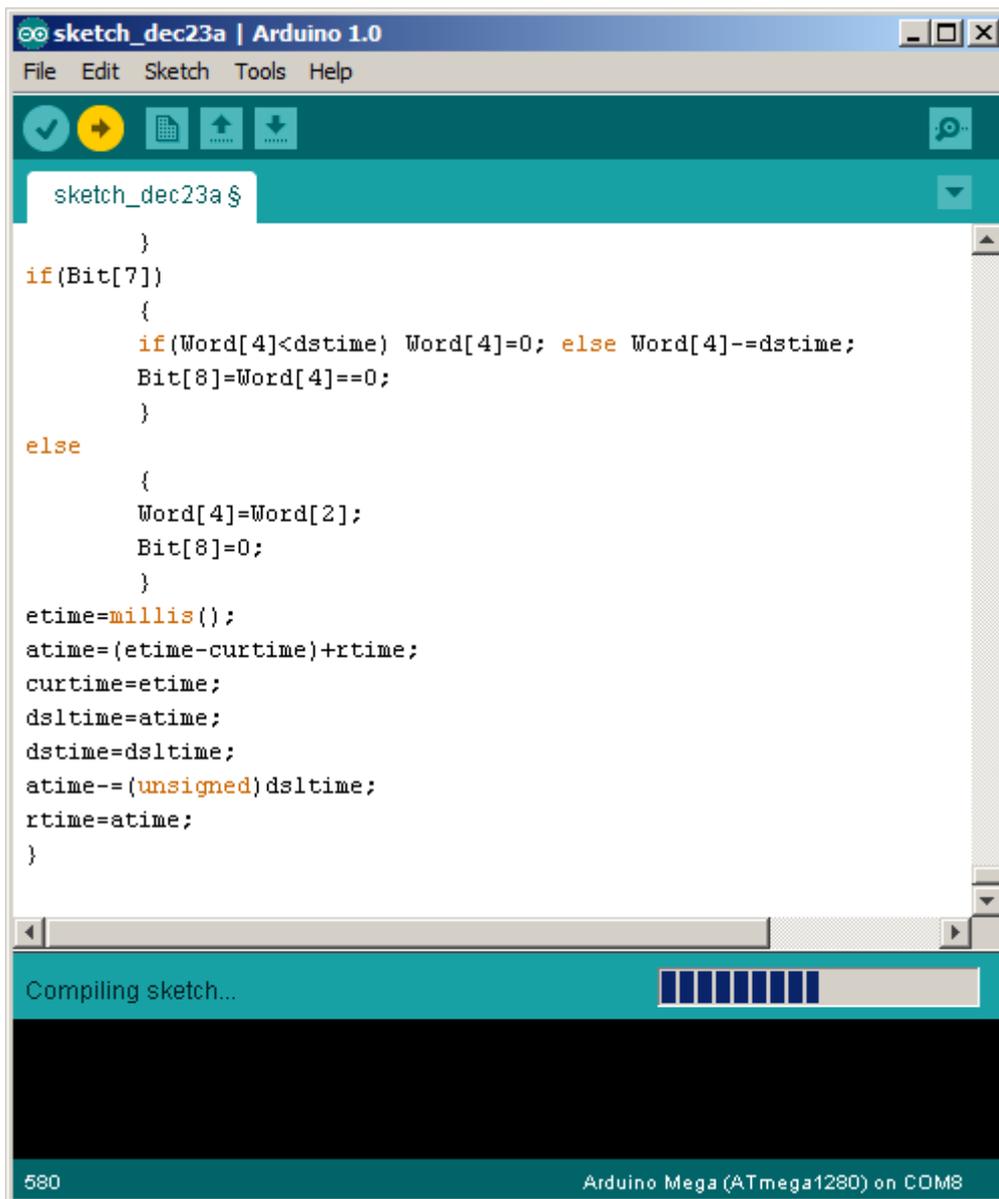
"arduinoloadw.exe" program, present in the AUTOMGEN installation directory is automatically launched at the end of compiling. This is done with the following option:

L'exécutable "arduinoloadw.exe" présent dans le répertoire d'AUTOMGEN est automatiquement lancé en fin de compilation en configuration cette option dans le projet AUTOMGEN :

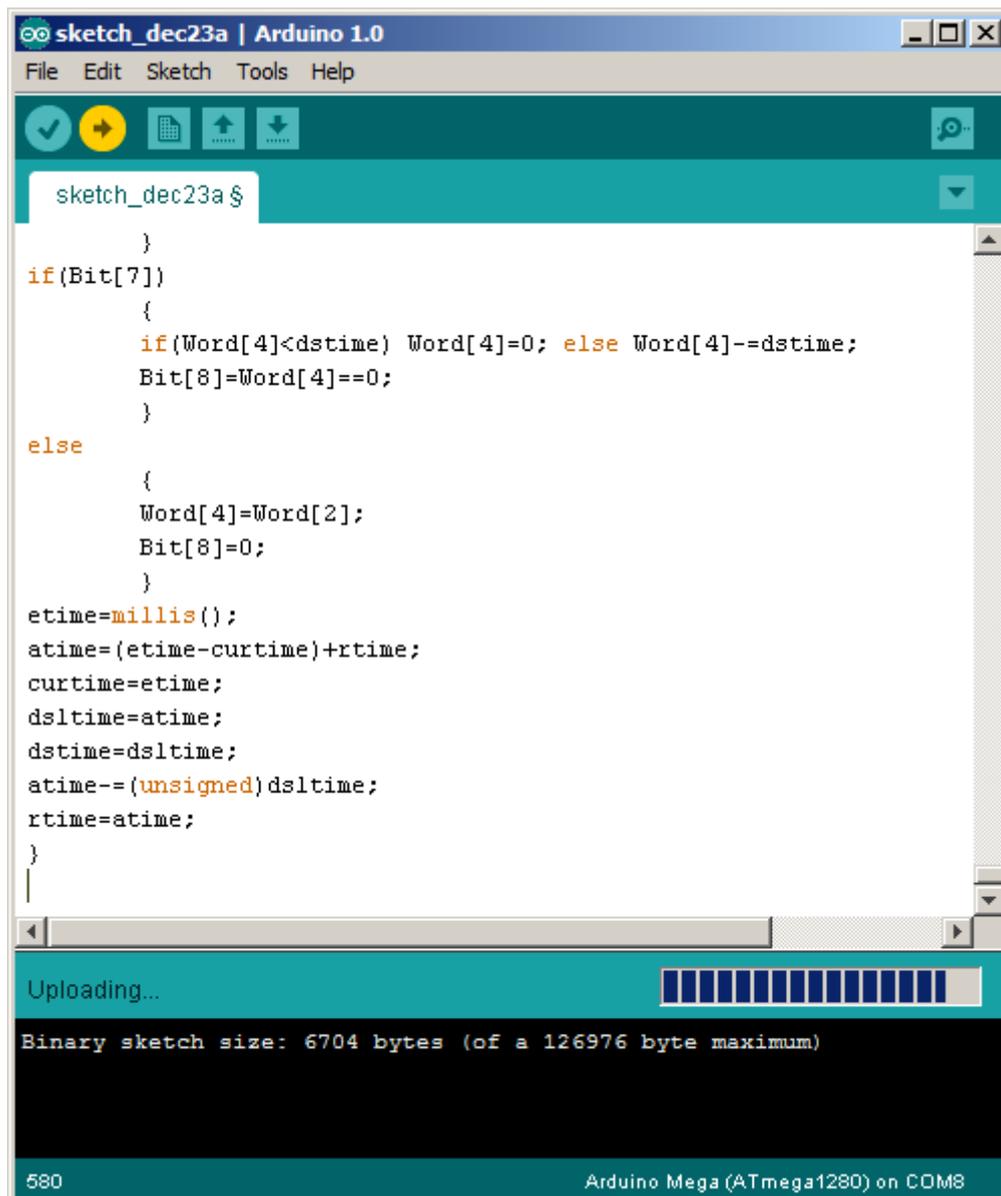


Arduino software must be launched before compiling. Board type and COM port have also to be selected in this software before compiling.

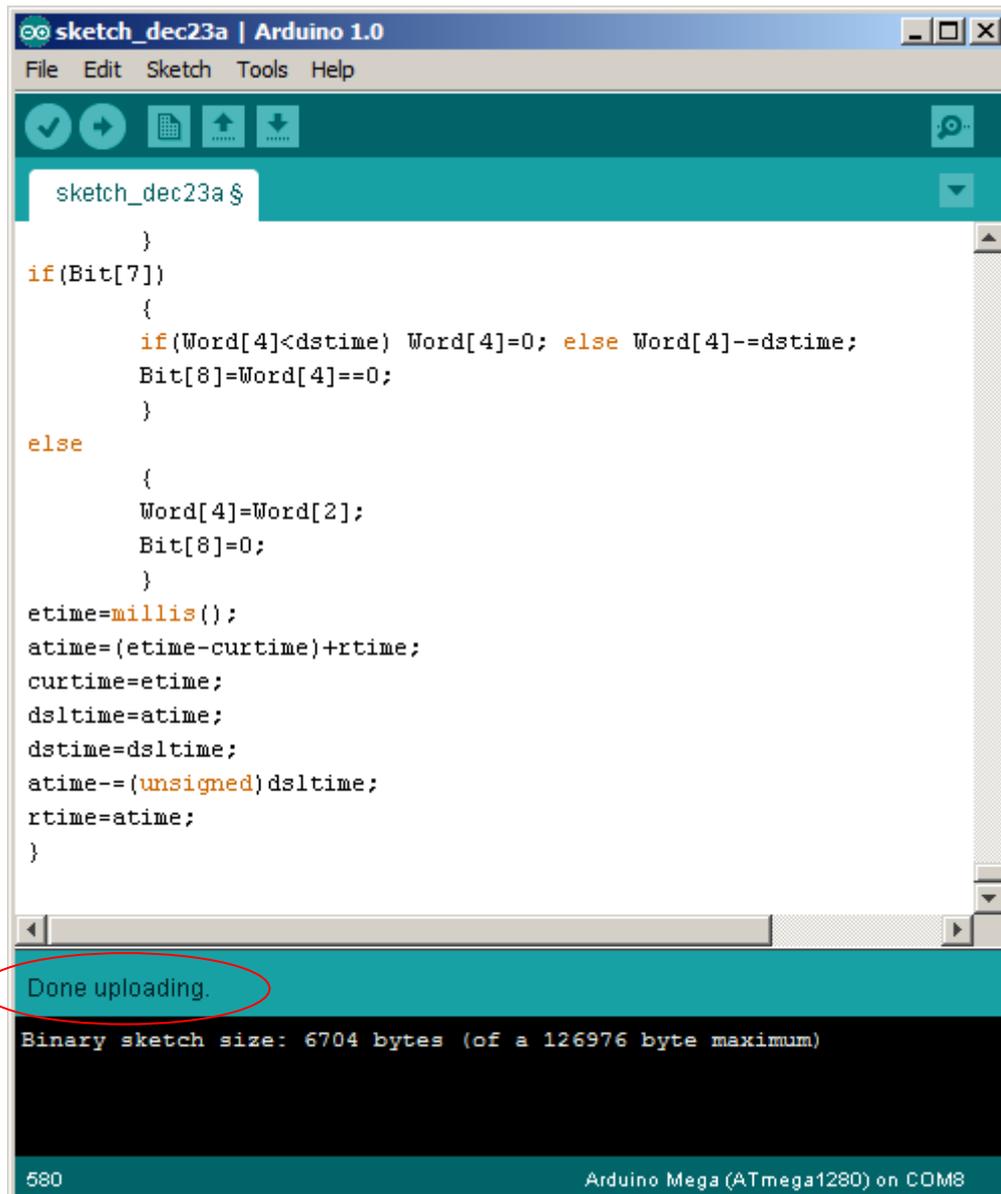
Click on GO or launch compilation in AUTOMGEN will process the following actions:



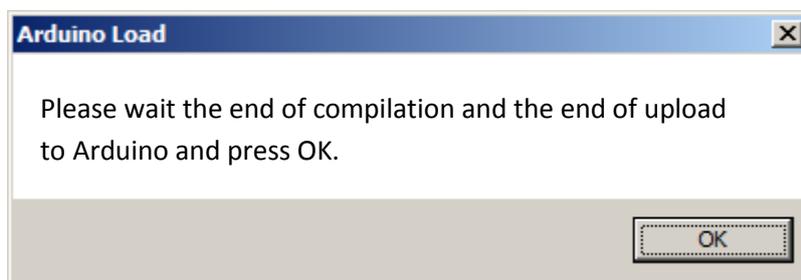
Compiling



Uploading



When "Done uploading" is displayed in Arduino software, click OK in the following message box:



If you have clicked on "GO" in AUTOMGEN, connect d mode will be automatically activated.

Notes :

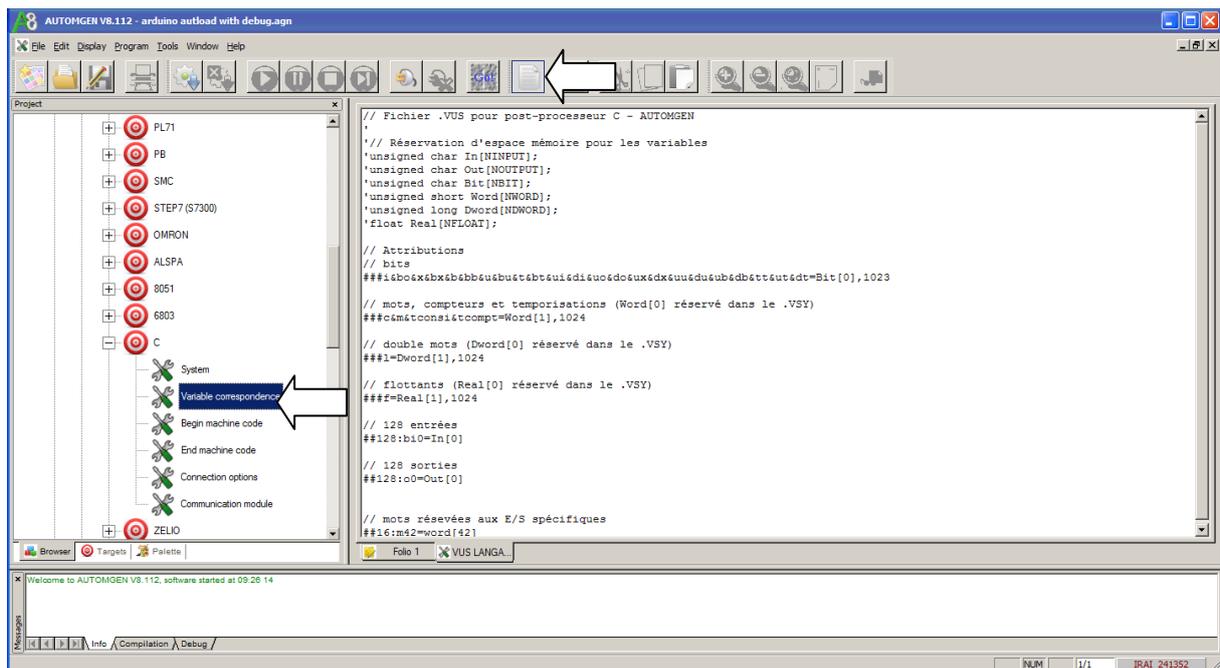
- no need to launch again Arduino software for the next compilations,
- only one Arduino software instance must be launched.

Mapping Automgen variables to Arduino I/O

The C post-processor used for generating code for Arduino uses variables array :

```
'unsigned char In[NINPUT];
'unsigned char Out[NOUTPUT];
'unsigned char Bit[NBIT];
'unsigned short Word[NWORD];
'unsigned long Dword[NDWORD];
```

These array can be seen in the Configuration/post-processor/C/variables correspondences items:



These correspondences define links between Automgen variables and the C variable arrays:

```
// bits
##i1=bits
##c&m&t&con&si&t&com&p&t=Word[1],1024
```

```
// words, counts and timers
##c&m&t&con&si&t&com&p&t=Word[1],1024
```

```
// double words
##l1=Dword[1],1024
```

```
// reals
##f=Real[1],1024
```

```
// inputs
##128:bi0=In[0]
```

```
// outputs
##128:o0=Out[0]
```

By example, i0 Automgen variable is associated to item #0 of the "In" array.

For more informations, please see the "variable correspondences" chapter of the Automgen manual.

The " Configuration/post-processor/C/begin machine code" item contains 2 functions for copying C variables array to the arduino physical I/O :

The getinputs function is used to read physical inputs and copy them into c variable array.

The setoutputs function is used write physical outputs from c variable array.

The setup function is used to set the I/Os (defininig I/O type by example).

Examples :

```
'void setup()
'{
'pinMode(13, OUTPUT);
'pinMode(14, OUTPUT);
...
}
```

Define ports 13 and 14 ad outputs

```
void getinputs(void)
{
'In[0]=digitalRead(22);
'In[1]=digitalRead(23);
}
```

Read physical input #22 and copy it into item #0 of the In array (input #0 of Automgen).

Read physical input #32 and copy it into item #1 of the In array (input #1 of Automgen).

```
void setoutputs(void)
{
'digitalWrite(13, Out[0]?HIGH:LOW);
'digitalWrite(14, Out[1]?HIGH:LOW);
}
```

Write physical output #13 from item #0 of the Out array (output #0 of Automgen).

Write physical output #14 from item #1 of the Out array (output #1 of Automgen).

The used functions (digitalWrite by example) are described in detail in the Arduino documentation.

Using numericals I/Os

Example for writing an analog output:

```
analogWrite(2, _m201_);
```

Write analog output #2 from Automgen word #201. The `_automgen` variable name `_syntax` allows to reference directly an Automgen variable. This is useful if we don't know where is located a word in the C variables array (case of the automatic mapping).

```
_m200_ = analogRead(A0);
```

Copy the value from analog input A0 to the Automgen word #200.

Examples can be found in the `examples\post-processor\C` sub directory of the Automgen installation directory.