Programming in BASCOM-AVR

The AVR is a powerful modern microcontroller produced by a company called Atmel http://www.atmel.com . BASCOM-AVR is a compiler that uses a version of Basic very similar to QBASIC to produce programs for the AVR. A free version BASCOM-AVR is available from MCS Electronics from their website at http://www.mcselec.com . BASCOM-AVR uses an Integrated Development Environment (IDE) which allows you to write and edit programs, compile them, test them with a simulator and finally write the program to the microcontroller for use in a circuit all from one program.

To use BASCOM-AVR open the applications folder on the desktop and double click BASCOM-AVR.

Set up the programmer
Go to Options -> Programmer and in the Programmer drop down menu select 'STK200/STK300 Programmer'.

Set up the chip
Go to Options -> Compiler -> Chip in the Chip drop down menu select '2313def.dat'
Select the Communication tab and in the Frequency drop down menu select '8000000'

Connecting the starter kit

Semiconductor manufacturers provide starter kits to make it easy for engineers to learn about the manufacturers parts to encourage them to use them in their designs. The STK200 starter kit is designed to help engineers quickly get used to using the AVR microcontroller. The kit is fitted with the Atmel 90S2313 AVR microcontroller.

Connect the starter kit to the parallel port using the lead provided. Connect the power lead to 12v from the power supply and turn on the power supply and the power switch on the STK200. The red on light should light.

Start writing a new program
Go to File -> New

Or pressing 

Program 1

There are two ports on the 90S2313 AVR microcontroller. Each pin on a port can be used as an input or an output. Information in the data direction register sets the direction, a 0 means input and a 1 means output. The LEDs on the STK200 are connected to port B has address 38H, the data direction register for port B is at address 37H. To use the LEDs we need to set all port B pins as outputs so send 11111111 to address 37H.

- Type this program into the edit window

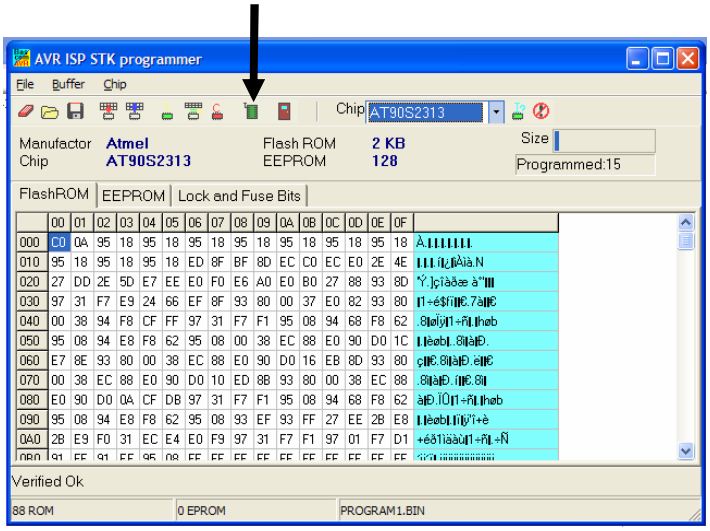
```
Rem Set up Port B as an output port
Out &h37, &hFF

Rem Send 01H to the output port
Out &h38, &h02
End
```

Now click on  to compile (turn Basic to machine code).

Then click on  to turn on the programmer.

This window should appear. Click on auto program chip. The ISP LED on the STK200 should flash.



A 0 lights up an LED and a 1 turns an LED off. What evidence do you have for this?

Program 2

- Change the program to make the middle two LEDs light up

Program 3

Variables have to be *declared* before they are used in BASCOM-AVR. Declaring a variable tells the compiler what kind of information the variable will hold so how much memory will be needed to store it. You use Dim *variable_name* As *variable_type* to declare a variable.

The variable types are:

Bit can hold only the value 0 or 1
Byte can hold a whole number 0 to 255
Integer (two bytes) are whole numbers from -32,768 to +32,767
Word (two bytes) are unsigned numbers ranging in value from 0 to 65535
Long (four bytes) are whole numbers with values from -2147483648 to 2147483647
Single (four bytes) are decimal numbers with values from 1.5 x 10^–45 to 3.4 x 10^38
String (up to 254 bytes) are a series of letters

- Type the following program into the edit window.

```
Rem Tell the compiler that J is a long integer
Dim J As Long

Rem Set up Port B as output port
Out &h37 , &hff


Do
  Out &h38 , &h01
  Gosub Waste
  Out &h38 , &h00
  Gosub Waste
Loop

Rem Subroutine to waste some time
Waste:
  For J = 1 To 100000
  Next J
Return
```

- Compile and auto program the chip

Program 4

Change the program to produce a set of lights: □□□■■□□□, □□■□□■□□, □■□□□□■□, ■□□□□□□■, □□□■■□□□, □□■□□■□□, …

Program 5

Type in this program, compile and load onto the chip. Try pressing the switches on the STK200

```
Dim B As Byte
Out &H37 , &HFF
Do
B = Inp(&H30)
Out &H38 , B
Loop
```

How does the program work?

When a button is pressed does it produce a 1 or a 0?

The AT90S2313 has 15 input/output pins, what effect does this have on how your program behaves?

Program 6

Try this program that uses polling

```
Dim L(4) As Byte
Dim J As Long
Dim S2 As Byte
Dim X As Byte
Out &H37 , &HFF

L(1) = &HE7
L(2) = &HDB
L(3) = &HBD
L(4) = &H7E

DO
For X=1 To 4
  Out &H38 , L(X)
  GOSUB delay
Next X
Do
S2=Inp(&h30)
S2=S2 And &h02
Loop Until S2 = 0
Loop
End
```

```
delay:
For J = 1 To 20000
Next J
Return
```

Which button do you press to make the LED sequence operate?

Program 7

Change the program so that it uses button 5 to make the sequence repeat

Program 8

Change the program to make it produce the pattern □□□■■□□□, □□■□□■□□, □■□□□□■□, ■□□□□□□■ when button 2 is pressed and ■□□□□□□■, □■□□□□■□, □□■□□■□□, □□□■■□□□ when button 3 is pressed.