int void main says that which port we can make output and which we can make input

while(1) says which port you can turn on and off you can turn on and turn off.

0b tell you its binary 0000000 (number of digits tell you number of ports)

DDRB means port b

DDRA means port a

and so on

now DDRB 0b 00000 \*means all 5 ports in b are input, when 1 is there it is output

now to make a certain pin high and low

we use while loop

while(1) // means 1 is true

{ PORTB = 0b 00110

alternative syntax is PORTB = (1 << 7); put 1 after 7 places hence it will become 10000000

Steps to make a board do something:

1) Write a program and tell the board what to do.

2) Compile: turn the program into machine code.

3) Using the programmer to transfer the machine code to the board. The programmer is like the flash card reading for retrieving or storing data. It does the hard work to verify and transform data.

4) Debug and test.

What is EEPROM?

Eletrically Erasable Programmable Read Only Memory. It is non-volatile in nature and don’t lose data on power cut off. Follow this link.

What is SRAM?

Static Random Access Memory. It is volatile in nature and they lose data content when the power is switched off.

Follow this link https://www.youtube.com/watch?v=kU2SsUUsftA

What is make and C file?

Make file contains a set of instructions to be executed.

C file contains the actual program that is to be dumped onto the microcontroller.

Flash storage is a data storage technology based on high-speed, electrically programmable memory.

The speed of flash storage is how got its name: It writes data and performs random I/O operations in a flash. Flash storage uses a type of nonvolatile memory called flash memory. Nonvolatile memory doesn’t require power to maintain the integrity of stored data, so even if your power goes out, you don’t lose your data. In other words, nonvolatile memory won’t “forget” the data it has stored when the disk is turned off.

The AVR chip runs whatever program is stored in the flash, uses the RAM for temporary storage and the EEPROM for longer term storage.

turning my board into arduino.

Two ways are there either use Arduino as ISP or either use USBtiny as ISP.

http://www.dewassoc.com/support/msdos/decimal\_hexadecimal.htm - hex to binary

using USBtiny as ISP.

1. I am using the PCB I made in the previous week for embedded programming this week. My PCB board uses ATtiny 44. I will be using USBtiny to

program the board using Arduino IDE. Arduino IDE doesn't have ATtiny 44 as one of the boards listed by deault.

Hence the first step to add the ATtiny 44 board in the boards list. The steps followed are as follows:

2. Open File>Preferences>Settings tabs. At the bottom, look for Additional Boards Manager URLs and add:

https://raw.githubusercontent.com/damellis/attiny/ide-1.6.x-boards-manager/package\_damellis\_attiny\_index.json and click ok.

3.Open the boards manager in the “Tools > Board” menu. Type "ATtiny" in the search box and you will see ATtiny entry and click "Install".

Now close the boards manager.

4. Now if you go to Tools>Board and swipe at the bottom, you will see ATtiny MCUs.

5. Go to Tools, select ATtiny 24/44/84 from boards, select ATtiny 44 from processor, select "External 20Mhz" from clock and port accordingly.

Lastly select USBtinyISP as programmer.

6. Before burning the bootloader, connect SPI header pins of USBtinyISP to your PCB board, which you want to programme. Connect USBtinyISP

to USB port of your PC.

7. Now click "Burn bootloader". You will see lot of things running amd see "Done burning bootloader".

8. Now to check whether your board has been bootloaded and can run a program, run a Blink program. I edited the Blink program and changed

the pin according to my board.

9. You will see this once the program has been uploaded.

http://highlowtech.org/?p=1706

http://highlowtech.org/?p=1695

Helpful links

http://www.ladyada.net/learn/avr/index.html

Explain about fuses

using Arduino as ISP.

1. We will follow step 1 as mentioned above for USBtiny as ISP. We will have to first turn Arduino into a ISP.

2. Go to File>Examples>ArduinoISP and select ArduinoISP sketch and connect your Arduino.(don't upload the sketch now)

3. Now got to Tools tab and select board as "Arduino Uno" and the COM to which you arduino is connected. Now upload the sketch. You should see something like this.

4. Next connect

Arduino Pin 10: RESET

Arduino Pin 11: MOSI

Arduino Pin 12: MISO

Arduino Pin 13: SCK

Arduino's GND to GND of 2x3 SPI header

Arduino's 5V to VCC of 2x3 SPI header

5. Now go to Tools again, select board as "ATtiny24/44/84", select processor as "ATtiny44", Clock "External 20MHz", Port as mentioned and Programmer as "Arduino as ISP".

6. Now click "Burn Bootloader" and you will see, something like this. You can now upload sketches from Arduino to your board.

7. Now to check whether your board has been bootloaded and can run a program, I ran the same Blink program again and it was uploaded successfullly.

Executing C from Arduino IDE

In ATtiny 44 there are two kind of ports, Port A or PA(8 in number) and Port B or PB(4 in number)

1. I wanted to see if I could run the same program using C. The layout of the program in Arduino IDE and C is kind of same, the only difference is between the syntaxes

used.

2. In Arduino IDE we define port for output as pinMode(PA0, OUTPUT); while in C we define it as DDRA = 0b00000001, where DDRA means(data direction register for port A), 0b means binary value and 00000001 means that pin 0 is output.

3. void loop() in Arduino IDE is replaced by while loop in C. digitalWrite(PA0, HIGH) in Arduino IDE is replaced by PORTA = 0b00000001.

4. delay() is replaced by \_delay\_ms()

5. I also realized that once you have written the programme in C, you can flash the program from Arduino IDE directly to your board.

Using ATmel Studio.

1. C programme can run directly on Atmel STudio. It can also import Arduino sketches. I downloaded the Atmel Studio 7.0 (build 1931) web installer. Please check all the

architecture while downloading. Download it from here.

2. Go to File>New Project and click okay

3. Select your device by typing "ATtiny" and select "ATtiny 44". It will open a new window where you can write the program in C.

4. No go to Tool>External Tools and add the following information

Debug Version

Title:- USBTiny ISP Debug

Command :- avrdude.exe

Arguments :- -c usbtiny -p attiny44 -U flash:w:$(ProjectDir)Debug\$(TargetName).hex:i

And click save.

5. You can write the code or just copy another code in C.

6.