LDI – Stands for Load Immediate. This loads an 8 bit constant directly to register 16 to 31.

SPL – Stack Pointer Low

SPH – Stack Pointer High

RAMEND – A label that represents the last memory address in SRAM. It’s a 16 bit word so we use the functions to split 16 bits into 8 bits so it can be handled.

Using low() and high(), we can return the low byte and high byte respectively of a 16-bit word

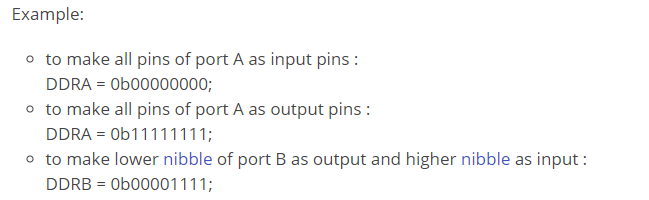
rol (Rotate left): The MSB is rotated to the carry flag, the carry flag is rotated to the lsb, all other bits are shifted left.

<http://www.elecrom.com/avr-tutorial-2-avr-input-output/>

Atmel AVR is 8 bit microcontroller. All its ports are 8 bits wide. Every port has 3 registers associated with each one of its 8 bits.

These three registers are as follows (x can be replaced by A,B,C,D as per the AVR you are using)

* DDRx register
* PORTx register
* PINx register

DDRx (Data Direction Register): It’s settings determine wether port pins will be used for input or output. Writing 0 to a bit in DDRx makes corresponding port pin as input, while writing 1 to a bit in DDRx makes corresponding port pin as output. 

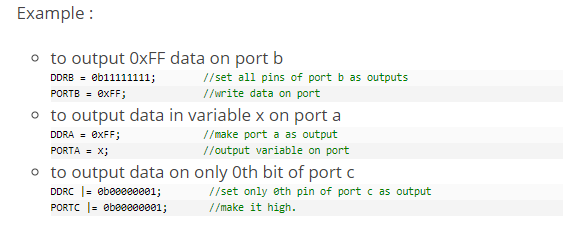
PINx (Port IN): Used to read data from port pins. In order to read the data from a port pin, first you have to change port’s data direction to input. This is done by setting bits in DDRx to zero. If port is made output, then reading PINx register will give you data that has been output on port pins.

PORTx is used for two purposes

1. To output data : when port is configured as output

When you set bits in DDRx to 1, corresponding pins become output pins. Now you can write data into respective bits in PORTx register. This will immediately change state of output pins according to data you have written.

In other words to output data on to port pins, you will have to write it into PORTx register. However do not forget to set data direction as output.



1. To activate/deactivate pull up resistors – when port is configured as input