

Aim of project to build a small computer which gcc can use as a target

- use the ATMega328p
 - 32K flash ram, 2K ram
 - cost: £1.95
 - [ATMega USB programmer](http://shop.tuxgraphics.org/electronic/detail_avrusb500smd2.html) (`http://shop.tuxgraphics.org/electronic/detail_avrusb500smd2.html`)
 - cost: £30.00

- other miscellaneous components, resisters, wire, prototype board
 - cost about £4.00

Why use ATmega processors?

- gcc supports them now and has for about one and a half decades
 - mature compiler
- harvard risc architecture
 - the above programmer can program nearly all of these devices

Atmel ATMega series of microprocessors

- the number of components to make a minimal system is tiny
- they are also extremely easy to interface to peripherals
 - A->D, D->A, pwm (servo and motor control) etc
 - multiple hardware timers etc

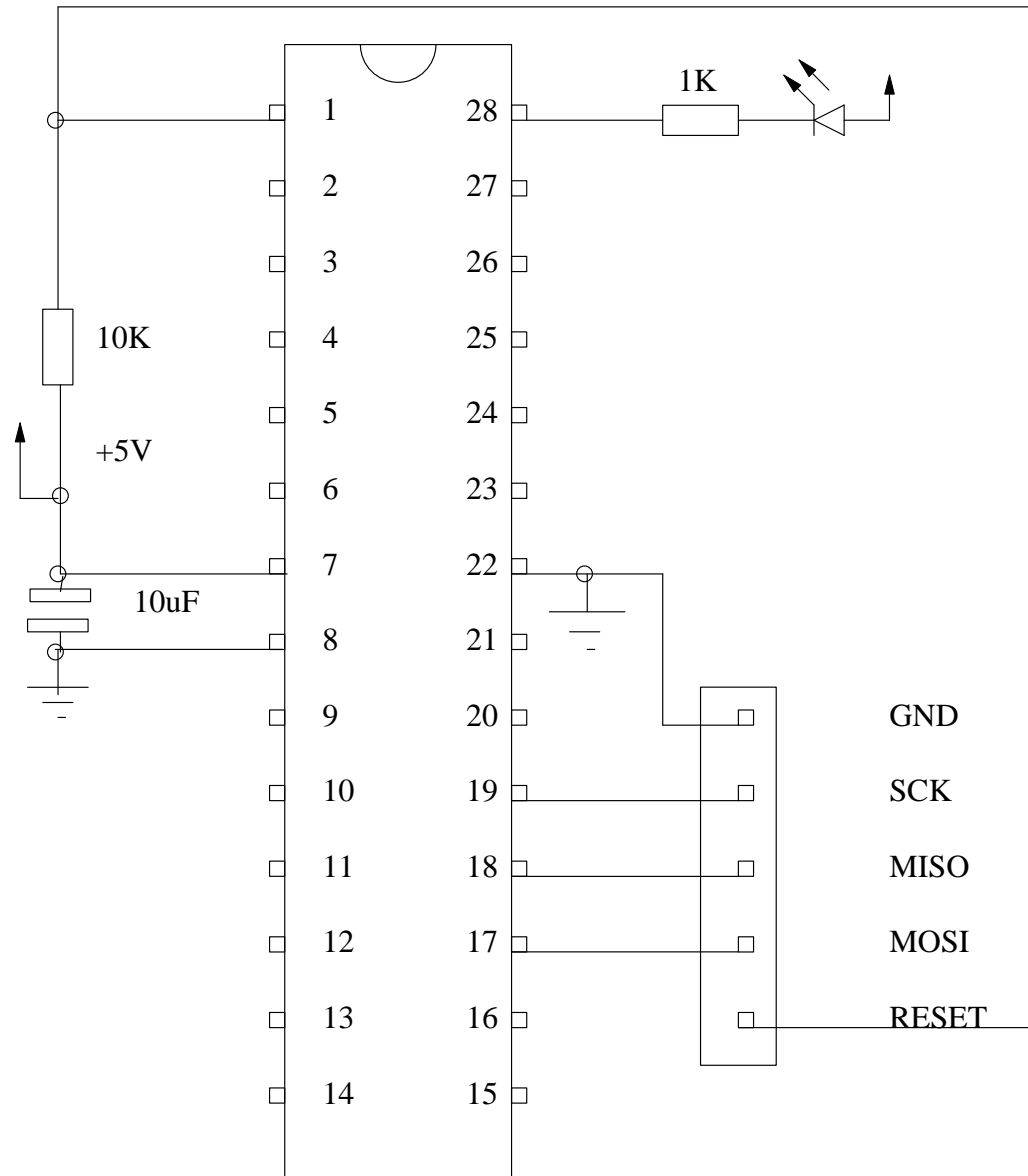
Atmel 328p

- is an 8-bit AVR RISC-based microcontroller (some of its features include):
 - 32KB flash memory
 - 1024B EEPROM, 2KB SRAM

- 23 general purpose I/O lines
 - 32 general purpose working registers
 - three flexible timer/counters with compare modes
 - internal/external interrupts, a 6-channel 10-bit A/D converter
 - programmable watchdog timer with internal oscillator

- many features omitted for sake of brevity

Simple computer flashing a LED with the ATmega328p



Packages to install under Debian or Ubuntu

- firstly ensure that floppsie.comp.glam.ac.uk is in the repository:

- `/etc/apt/sources.list`

- ```
...

GNU Modula-2 repo

deb http://floppsie.comp.glam.ac.uk/debian/ wheezy main
deb-src http://floppsie.comp.glam.ac.uk/debian/ wheezy main
...
```

# Packages to install under Debian or Ubuntu



```
$ sudo apt-get update
$ sudo apt-get install avrdude gcc-avr gm2-avr
```

## Configure avrdude config file /etc/avrdude.conf

```
default_serial = "/dev/ttyUSB0";

programmer
 id = "avrusb500";
 desc = "tuxgraphics avrusb500";
 type = stk500v2;
;
```

- this ensures that the TuxGraphics programmer can be seen by avrdude
  - double check that this entry is not already in /etc/avrdude.conf



# Build Computer!

- **build circuit** `<http://floppsie.comp.glam.ac.uk/download/avi/Minimal_ATmega328p.vob>`

## Build Computer!

- parts can be found from many sources, eg: [from a kit](http://shop.tuxgraphics.org/electronic/detail_dotmatrix-board.html) `<http://shop.tuxgraphics.org/electronic/detail_dotmatrix-board.html>`

## Write flashing LED program: `flashled.mod`

```
MODULE flashled ;

PROCEDURE Turn (on: BOOLEAN) ;
BEGIN
 IF on
 THEN
 (* turn LED on *)
 ASM VOLATILE ("cbi 8,5");
 ELSE
 (* turn LED off *)
 ASM VOLATILE ("sbi 8,5");
 END
END Turn ;

(*
 InitLed - initialize pin 0 as an output
*)

PROCEDURE InitLed ;
BEGIN
 ASM VOLATILE ("sbi 7,5")
END InitLed ;
```

## Write flashing LED program: `flashled.mod`

```
CONST
 Delay = 400 ;

VAR
 i, j: CARDINAL ;
```

## Write flashing LED program: `flashled.mod`

```
BEGIN
 InitLed ;
 Turn(FALSE) ;
 LOOP
 FOR i := 0 TO Delay DO
 FOR j := 0 TO Delay DO
 ASM VOLATILE ("nop")
 END
 END ;
 Turn(TRUE) ;
 FOR i := 0 TO Delay DO
 FOR j := 0 TO Delay DO
 ASM VOLATILE ("nop")
 END
 END ;
 Turn(FALSE)
 END
END flashled.
```

## Compile and link code



```
$ avr-gcc -mmcu=atmega328 flashled.mod -flibs=min -o flashled.elf
$ avr-objcopy -R .eeprom -O ihex flashled.elf flashled.hex
```

## Link up USB programmer to circuit

```
$ sudo avrdude -p m328p -U lfuse:w:0xe2:m -U hfuse:w:0xda:m \
-U efuse:w:0x05:m -c stk500v2 -e -v -U flash:w:flashled.hex
```

## Link up USB programmer to circuit

- watch out for microprocessor part numbers, there exist 328p, 328p-pu
  - which are nearly identical, the (328p-pu) has newer power saving internal circuitry and critically a different device signature
  
- you will need to modify `/etc/avrdude.conf` to reflect this
  - copy the 328p section and change the signature (ymm!)