ATmega328P Programmer Hardware for the Raspberry Pi

The Raspberry Pi makes an ideal development platform for the Atmel ATmega328P micro-controller.

There are two aspects to this platform:

1. The software as described at: www.noblemicro.com/Using_the_Raspberry_Pi_to_program_the_ATmega328P_AVR.pdf
2. The programmer hardware that is described below.

The basis of the programmer is an implementation of SPI (Serial Peripheral Interface) so that the Raspberry Pi and the AVR chip being programmed can communicate with each other.

The SPI interface defines four connections. There is a clock (SCLK), a chip select (SS) and separate transmit and receive data lines (MOSI and MISO).

These signal lines are available on both the Raspberry Pi and the ATmega328P. However, it is not necessary to connect the chip select (SS) since the configuration is for a master and single slave.

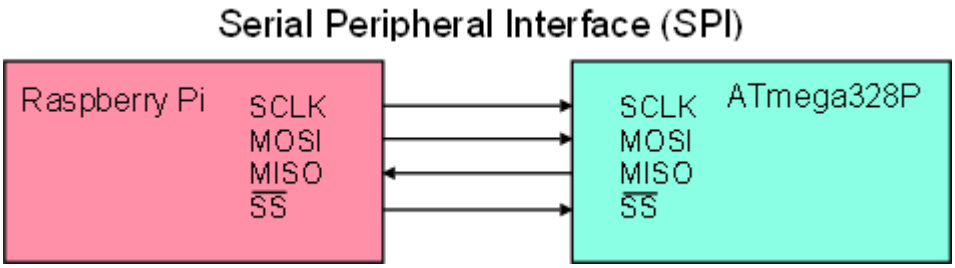
In addition, the ATmega has to be powered from the Raspberry Pi, necessitating two further wires for 3.3 volts and Ground.

Note that it is absolutely essential for the ATmega to be powered from the Pi's 3.3 volt supply and not from a separate 5 volt supply. The latter arrangement would cause irreparable damage to the Raspberry Pi which cannot tolerate 5 volt levels on its GPIO (General Purpose Input/Output) pins.

The ATmega requires its own external oscillator and so a 16MHz crystal and associated 20pF capacitors have to be included.

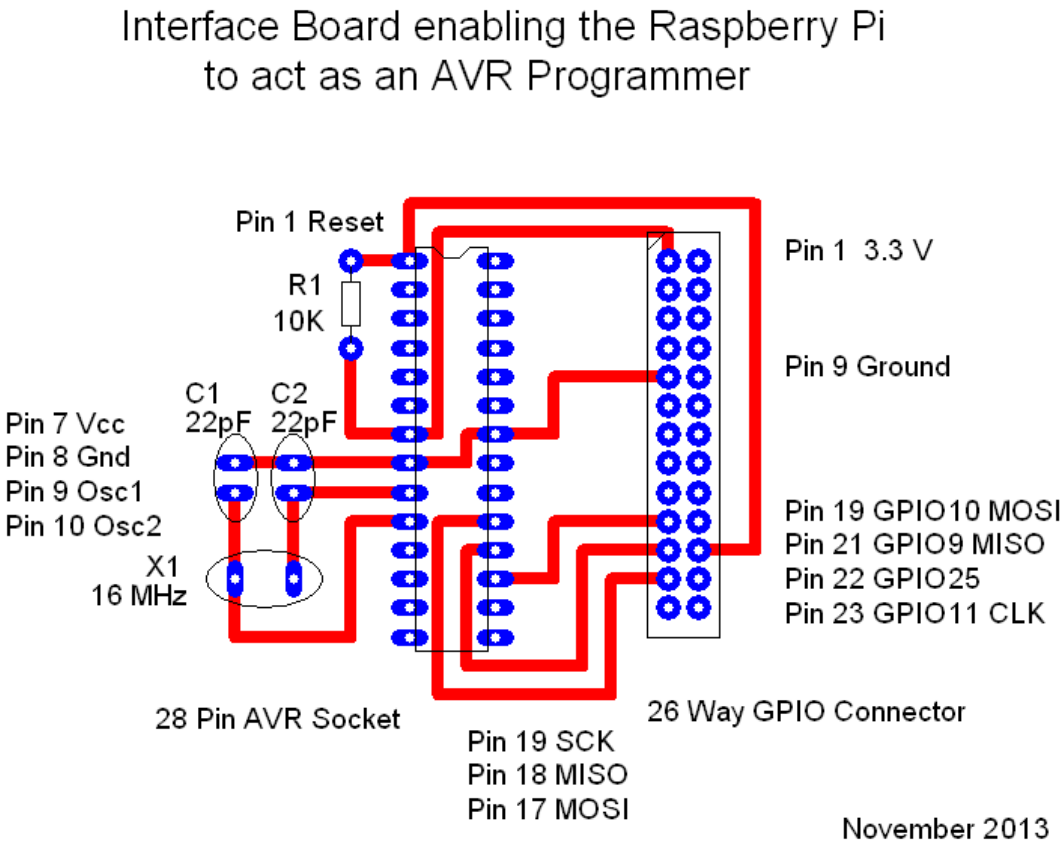
Finally, the programming process requires the Reset pin of the ATmega to be toggled and this is done using another digital output line from the Pi in conjunction with a pull-up resistor.

The SPI connection logic is shown here:

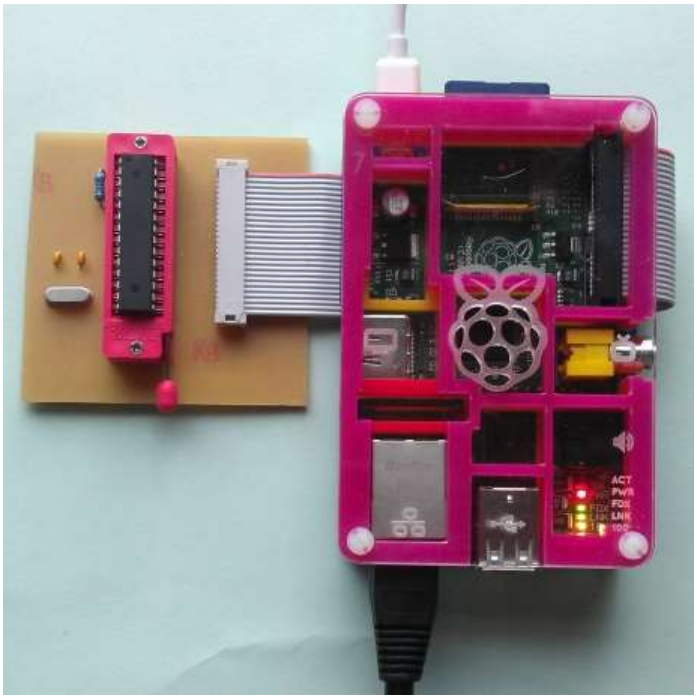


The circuit layout for connecting the ATmega to the GPIO connector of the Raspberry Pi is shown below. Note that it is not necessary to connect the SS line as the Pi, acting as Master, is only addressing one Slave.

GPIO pin 22 (Output signal 25) is used to control the Reset pin of the ATmega



The above scheme has been implemented as a custom printed circuit board (shown below) using a ZIF (Zero Insertion Force) socket for the ATmega.



The scheme can also be implemented using a breadboard. This requires the chip to be plugged directly into the breadboard rather than by using a ZIF socket.

The picture below shows the breadboard implementation. The connecting wires are marketed as “single-pin” cables and have a pin connection at one end (for the breadboard) and a socket connection at the other end (for the Raspberry Pi GPIO header).

The final picture gives a close-up of how the jumpers are interconnected.