EE459 Lab Assignment 2
Get the Microcontroller Working

1 Introduction

The purpose of this assignment is to install the microcontroller and associated hardware on the project board and to get it working. Along with the microcontroller and clock oscillator, an LED will be installed and connected to one of the micro's I/O pins. Software will be written to make the LED flash on and off at a specified rate. This is meant to be the simplest possible task to make the micro do, so there is no point in trying to make your project do anything else until it can do this.

2 Components

Each team will be provided with the following components:

- ATmega328P microcontroller
- Two 16-pin sockets for the microcontroller
- 7.3728MHz oscillator
- 14-pin socket for oscillator (or a 16-pin socket with two pins cut off)
- 0.01μF capacitor
- LED and 270Ω resistor
- Two 3-pin headers for the programming connector
- 1kΩ resistor

3 Install the Clock Oscillator

Spend a few minutes deciding where the oscillator and microcontroller should be installed. They should be near each other and probably close to where the power connections are on the board in order to keep as much of the rest of the board free for other components. As with the other components, don't install them out in the middle of the board since that space will be need for various project components.

1. Your parts should include a 14-pin socket. Install this socket for the oscillator by soldering at least two of the corner pins to the metal pads. The socket is marked in some way to show which is pin 1.
2. The connections for the oscillator are shown in Figure. 1. Wire power to pin 14 and connect pin 7 to ground.
3. Install the 0.01μF capacitor close the oscillator and wire it between the power and ground pins on the oscillator socket (14 and 7) as shown in Fig. 2.

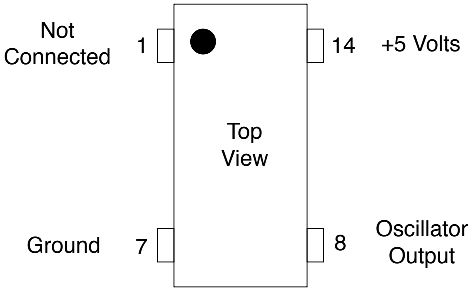

Figure 1: Pinouts for Oscillators

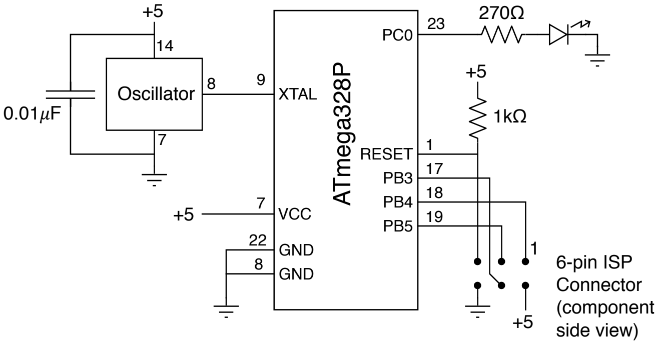

Figure 2: Wiring diagram for ATmega328P

4. Insert the oscillator chip into the socket making sure to orient it correctly. Depending on which type of oscillator you have, pin 1 on the oscillator may be marked by a black dot or a some other marking that is not present by the other 3 pins.

You are now ready to apply power to the board and check the clock signal with one of the oscilloscopes.

Turn on the Tektronix DPO2014 or MSO2014 oscilloscope on the bench. From the black pouch behind the scope remove one of the probes and connect it to channel 1. The probes have colored markers on them that are used to identify which probe is connected to which channel so use the probe with the color that matches that of channel 1.

Important: For all measurements with the scopes use one of the scope probes. Do not use one of the cables that has a BNC connector on one end and two clip leads on the other since these will give significantly worse performance in showing ringing on the signal.

Oscilloscopes usually use either a “1X” or “10X” probe (which some manufacturers call “X1” or “X10” probes.) The 1X probe does not attenuate (reduce) the signal so a 4V signal in the circuit shows up at the scope as 4V. An 10X probe attenuates the signal by a factor of 10 so the same 4V signal in the circuit is 0.4V at the scope. However the attenuation also reduces the capacitance of the probe which reduces ringing and other undesirable effects when measuring signals. For most measurements, using an 10X probe will give better results. The Tektronix scopes in the lab come with either a fixed 10X probe or one that can be switched between 1X and 10X. There should be a label on the connector of the probe that is attached to the scope that says “1X/10X” or just “10X”. Probes that can be either 1X or 10X have a switch on the barrel of the probe labeled “1X” and “10X”. For these probes slide the switch to the 10X position.

The scopes have no way of knowing whether the probe attached to one of the four channels is an 1X or 10X probe so you must change a setting that tells the scope what you are using in order for the correct values to be shown along the vertical axis. Press the colored button for the channel you are using and it will put up a display of channel settings along the bottom of the screen (Fig. 3). If the “Probe Setup” label doesn’t say “10 X”, press the key below that label and then press the “Set to 10 X” button along the right side.

Connect the probe tip to pin 8 of the oscillator chip and the probe ground to a ground connection somewhere on your board. Apply power to the board and use the scope to confirm that the output on pin 8 is a square wave of the correct frequency. The easiest way to get the scope to display your signal is to press the “Autoset” button near the right side of the scope. The scope will look for an input signal and configure its settings to display it. If the clock oscillator is working, you should get a display like that in that in Fig. 4.

There will likely be a lot of overshoot and ringing on the transitions from the zero to one and one to zero states so the output will not be a clean square wave. The main things to confirm is that the frequency is about right with a period of about $\frac{1}{7.3728MHz} = 135ns$, and that the high and low output levels look reasonable. The output in the high state should be at least 3V.

If the oscillator signal is about the right frequency but the signal doesn’t look anything like a squarewave, with the signal transitions rounded off and lots of other distortion (see Fig. 5), check that the 0.01μF capacitor

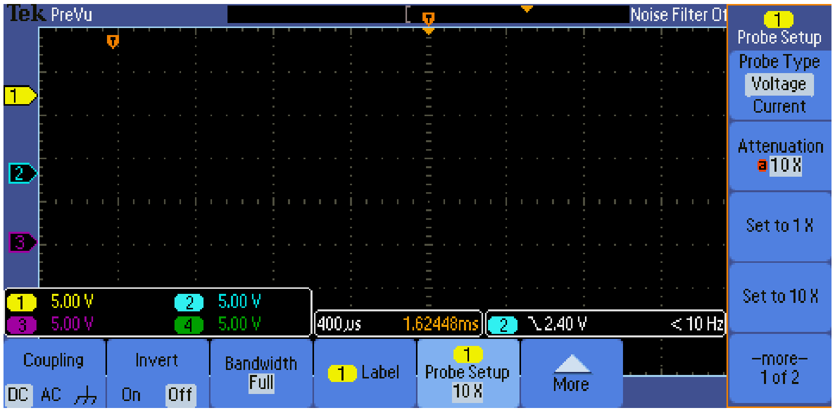

Figure 3: Scope display of channel settings

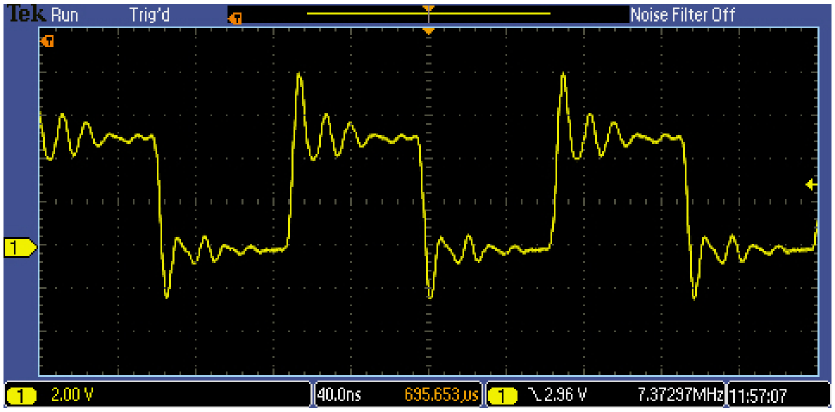

Figure 4: Scope display of the clock signal

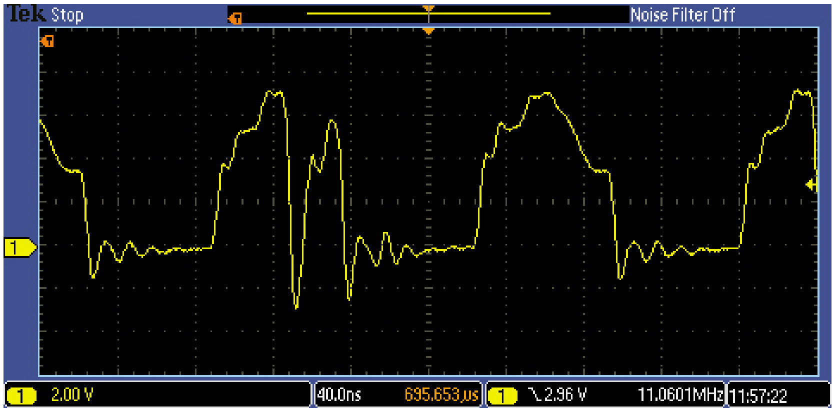

Figure 5: Scope display of the bad clock signal

has been correctly installed. Without the capacitor the signal shape is usually badly distorted.

The yellow marker along the left side indicates where ground or zero volts is located, and the low state of the signal should be close to this level. The numbers at the left side of the bottom of the scope display indicate that the vertical scale is 2 Volts/division. From this check the voltage level of the signal when it is in the high state. It should be close to 5 Volts although anything over about 3 Volts is probably OK. The number in the middle part of the bottom of the scope display indicates the horizontal time base or how much time each grid line represents (40ns in Fig. 4). Use this number to measure the period of the displayed signal and from that calculate the frequency. The scope may also directly calculate the signal's frequency and display it somewhere at bottom of the screen, and it should agree with the rated frequency of the oscillator.

4 Install the Microcontroller

Normally the microcontroller would be installed in a 28-pin socket. To save costs, you will be using two 16-pin sockets that can be mounted next to each other to form a 32-pin socket. One of the two sockets may have had four of the pins removed at the end of one of them. If so this one should installed so the empty holes are away from the other socket.

1. Install the sockets for the microcontroller by soldering two of the corner pins of each socket to the metal pads. It's not necessary to solder all the pins to the pads. Just two corner pins soldered will keep the socket from moving around.

2. The holes in the socket tend to be very tight when the sockets are new and this can make plugging the microcontroller into the socket very difficult. The holes can be loosen up slightly by inserting a piece of wire into the holes a couple of times. Find a piece of wire like the 22-gauge hookup wire available in the lab, or you can use one of the leads on a resistor, capacitor or LED. Using the wire, go along the socket and push the wire into each of the holes a couple of times to try loosening the contacts inside the socket.

3. Plug the microcontroller into the socket. It should seat fully to where there is no gap between the body of the IC and the top of the socket.

4. Using the wire-wrapping tool, make the wiring connections to the microcontroller socket for power, ground and the clock as shown in the schematic in Fig. 2.

5. Add a 1kΩ resistor as shown to pull up the RESET line.

5 Install the Programming Connector

The next step is to install the ISP connector for programming the microcontroller. The ISP connector is made using two 3-pin header strips that are installed next to each other so they form a 2×3 block of header pins (see Fig. 6). These can go anywhere on the board but should probably be near the microcontroller. The header strips have pins that stick out on both sides of the plastic part that holds it together and the pins are not the same length on both sides of the plastic. **The plastic part and the shorter pin should go on the component side of the board. The longer pin should go through the hole and stick out on the wiring side.**

Before soldering any of the pins to the board take your AVR programming modules and check that you will be able to press the black 6-pin connector down over the pins when they are installed in the position and orientation you are considering. Take a look at the small black connector on the end of the ribbon cable coming from the programming module. One of the holes that mates with the pins is marked with a small triangle on the connector. This indicates which of the six pins is pin number 1 on the connector and whichever of your pins it connects to will be pin 1 in your wiring. It's recommended that you make a mark on the component side of the board to indicate which pin is pin 1. Once you are sure the headers are in an acceptable position solder them to the wiring side of the board.

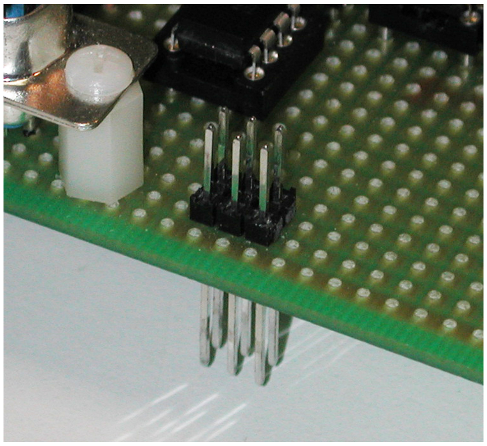

Figure 6: 6-pin ISP header mounted on a project board

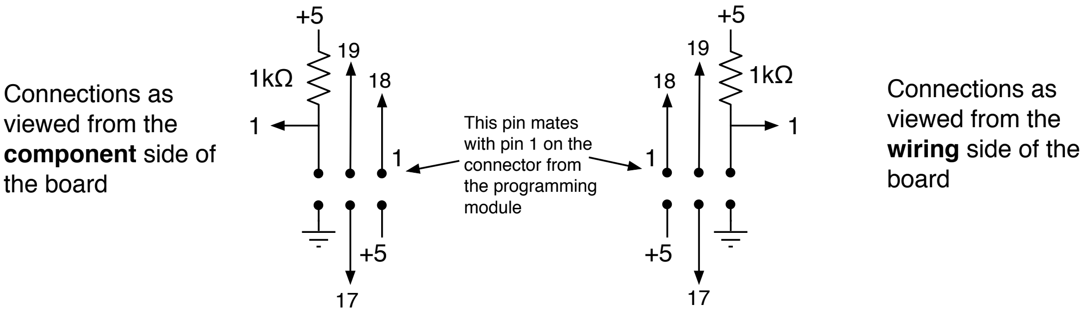
Connections as viewed from the **component** side of the board
+5
1kΩ
19
18
1
1
17
+5
This pin mates with pin 1 on the connector from the programming module
+5
19
1kΩ
18
1
1
+5
17
Connections as viewed from the **wiring** side of the board

Figure 7: Layout of ISP pins from both the component and wiring sides of the board

Tip on soldering the headers: In order to to make sure the headers are pressed fully agains the board and are also in straight, take the 6-pin connector of the AVR programming module and insert it over two of the pins at the ends of the rows so it's over one pin from each row. Use the connector to hold the pins in place while soldering the two pins at the other end of the rows. **Don't solder the pins that are inserted into the connector since this could damage the connector.** Once two pins (one on each row) have been soldered remove the connector and solder the other pins. All of the six pins should be soldered to the board so they can't move when the connector is pushed onto them.

Wire the six pins to the microcontroller according Fig. 2. Figure 7 shows the how the pins are laid out when viewed from both the component side of the board and from the wiring side. Make sure you are following the correct figure when doing the wiring. When looking at the pins on the **component** side, pin one should be in the upper right corner so it will mate correctly with the connector on the programming module. For more details on wiring the programming connector see the section on the "SPI Programming Interface" in the class handout on the ATmega328P microcontroller.

Note: Wiring the ISP connector incorrectly is probably the most common mistake teams make that prevents the microcontroller from working. Most of the time this is due to confusion about which is pin 1 on the connector and whether to wire according to the component side or wiring side view. If you have questions about this please see the instructor before doing a lot of wiring.

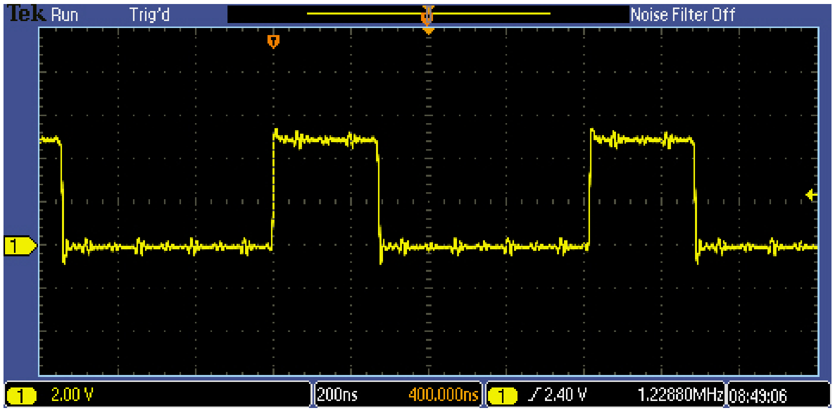

Figure 8: Scope display of the fast oscillation output

6 Fast Oscillation Test

At this point you are ready to test the micro with a simple program. Follow the instruction in the handout for the ATmega328P to program it with the sample program `at328-0.c` on the class web site that loops turning I/O port bit PC0 on and off as fast as possible. You should use the `Makefile` from the web site but it will probably have to be modified to work with your program and oscillator.

- The object file listed on the `OBJECTS` line must match the one for your program
- The `CLOCK` line must match the frequency of the oscillator provided

Note: The `Makefile` used in EE459 for programming the ATmega328P are different from ones use to program Arduinos as in EE109. Don't try to use a `Makefile` from an EE109 assignment to program the ATmega328P in EE459.

If this is the first time you have programmed this microcontroller chip, you will also have to do the "`make fuse`" command to program the configuration fuses. This step can be done right after doing the "`make flash`" command to download the program.

Once the micro is programed use the oscilloscope to observe the output. Connect the scope to PC0 (pin 23) and you should see the output signal going up and down as shown in Fig. 8. As mentioned before, use a real scope probe, not one of the test leads that has a BNC connector on one end and clips on the other. With a 7.3728MHz clock, the output of the ATmega328P should oscillate with each period having the output in the high state for about 270ns and in the low state for about 540ns.

Note: If your oscillation output is much slower than the above numbers it's probably because you didn't do the "`make fuse`" step above. If this wasn't done the microcontroller runs about 8 times slower.

7 Slow Oscillation Test

Install the LED and current limiting resistor somewhere on the board and wire them to pin 23 of the microcontroller as shown in Fig. 2. These will only be used for the first lab assignments so it isn't too important where you put them. The cathode of the LED is marked by a flat spot on the edge of the LED and this side should go towards ground.

With the LED installed, modify the program to add some delay to the loop so the output stays in both states for a longer period of time. Refer to the sample programs `at328-2.c` for ways to implement a delay in the program. Make the delay be 0.5 seconds in each state so the LED is flashing at a 1Hz rate.