Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science
6.115 Microprocessor Project Laboratory Spring 2024

Laboratory 1 INTRODUCTION
READ THIS FIRST to understand your hardware!

Issued: February 6, 2024 Due: Read it this week!

GOALS: Review SAFETY rules
Get familiar with your R31JP and PSoC development systems
Begin to get comfortable with the hardware in your kit
Install useful software packages for communicating with your microcontrollers
Understand the challenges associated with building real, performance circuits
Setup your communication software

READ: Scherz: "Grounds" (40-49 in 4th ed.); "Oscilloscope" and "Probe Suggestions" (598-607 in 4th)

We are issuing you more than the normal amount of equipment this term, including, for example, the loan of an oscilloscope. **Always be ready to work at home if the need arises.** Please do not travel with your kit far off campus, but do keep your kit with you, not stuck in your locker, for use in your living group if needed.

We (Skynet ☺) will issue you a lot of equipment, including three different microcontroller boards, a fantastic "briefcase-style" lab kit with breadboard space and power supplies and many useful test and measurement features, a collection of components including resistors, capacitors, and integrated circuits, an awesome oscilloscope, and a multimeter. Please familiarize yourself with the test equipment – play with the oscilloscope and the multimeter. We assume you have seen these before, but ask questions if you have not seen them. The oscilloscope is your light saber; the probe(s) should fly into your hand when you close your eyes. Always have it on and ready when you are building and debugging.

Do NOT damage your gear. The equipment costs a small fortune and is hand-made in many cases, e.g., The Blackbird kit. There are limits on the available equipment, and we may not able to replace or fix items with you quickly or, in some cases, at all. So this document is occasionally "direct" in language to make sure you are following. We want you to have working gear and to enjoy using it. It's your job to make sure that you use your gear carefully. Damage is not an excuse for missing a lab exercise or lab. ASK if you have any doubts or questions, we are excited to talk to you about the kit and gear. We have spent a lot of time building and assembling everything. The kit is filled with tools for having fun if you are thoughtful and careful.

The purpose of this document is to help you appreciate the gear, use it wisely, and point out areas that may raise questions in your mind. Do NOT start using the gear until you have attended the kit familiarization lecture in class, and ASK questions if you have any confusion.

SAFETY:

You have already read and signed the EECS department safety brief. Additionally:

If and when you work with your kit outside of the 38-600 laboratory, it CONTINUES to be an imperative that your safety and that of your family, dwelling, neighbors, loved ones and friends is the absolute priority for ANY and ALL work that you do for this class, same as in the 38-600 laboratory. "The Rules," below, therefore, are necessarily proscriptive. Use common sense and your safety familiarization. **For both 38-600 and at home**:

We insist:

1. Absolutely NO machining or power tools.
2. Absolutely NO wiring while the kit power is "on".
3. Absolutely NO voltages over 15 volts except when directed by a staff member in person.
4. Absolutely NO lithium or rechargeable batteries of any type.
5. Absolutely NO soldering or high heat sources.
6. Absolutely NO high currents (defined as in excess of 1 amp, as available from a USB port).
7. Absolutely NO high speed mechanical components in your project - no nail guns, quad-copters, etc.
8. Absolutely NO lasers, hot plates, radio stations, or any other energy sources that could be harmful.
9. Absolutely NO chemicals, foods, or biological materials involved in the project.

These points are NOT negotiable. If in doubt, do NOT do it. You are welcome to ask Professor Leeb any questions, but pleading will not change these rules. We hope that makes sense – we are concerned for your well-being and your enjoyment of the course material.

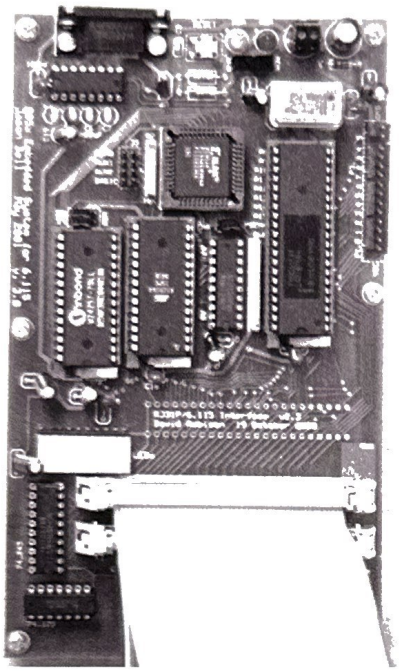
MICROCONTROLLERS:

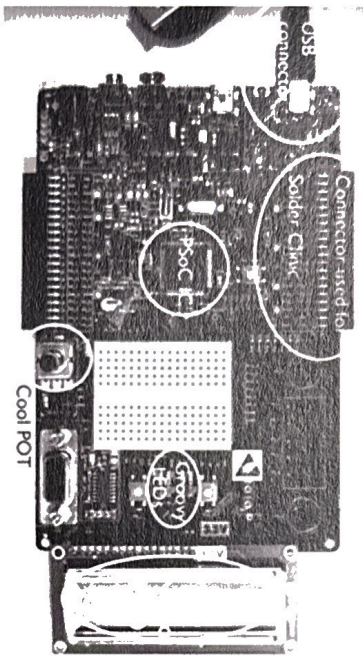
Your kit has two different microcontrollers available to you on three printed circuit boards, a wealth of exciting possibilities. One microcontroller, from the Intel 8051 family, is available on your R31JP microcontroller board, which connects to your briefcase kit. You have two copies of the other microcontroller, a Cypress 5LP PSoC (programmable system on chip), available on two different boards, a "Big Board" and a "Stick". The next page shows you pictures of the microcontroller boards, a block diagram of the "internals" of the two micros available on the boards (don't worry if you can't read every detail in the block diagram, we will discuss these more and with bigger pictures in class), and the Terminator most closely associated with each microcontroller's capabilities. Please look at the pictures on the next page very carefully, make sure you know which board is which:

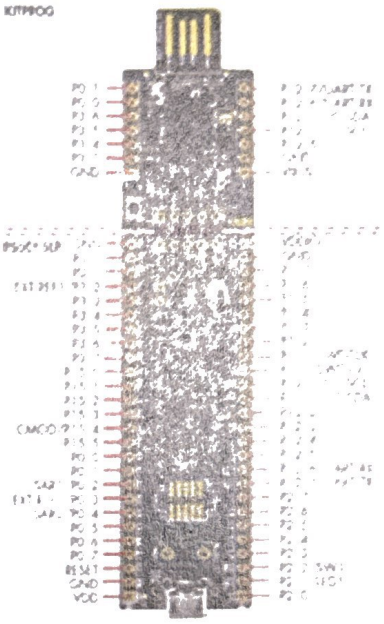
8051/R31JP

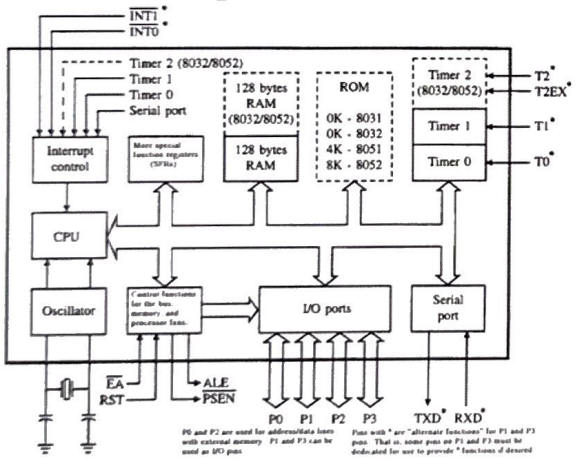
PSoC 5LP "Big Board"

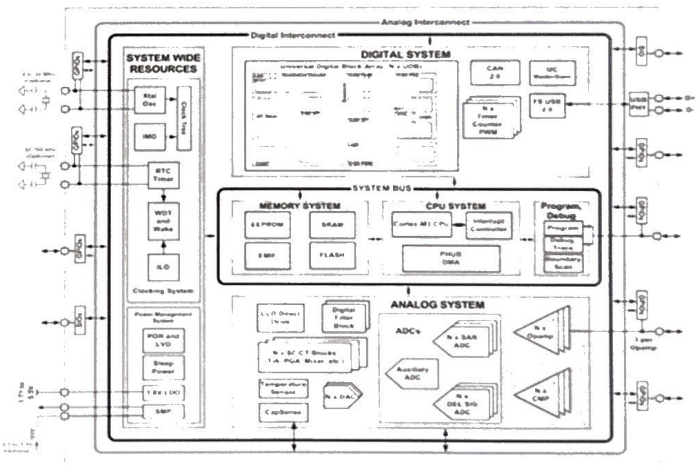
PSoC 5LP "Stick"





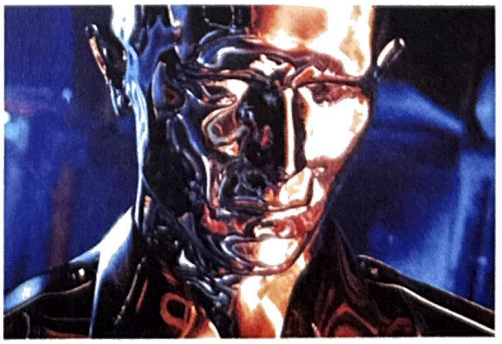






Cyberdyne 101: T800

T1000



The **R31JP** is built around an Intel 8051-family microprocessor, the largest IC on the board, a 40-pin plastic DIP (dual in-line pin) package. This board is a critical learning tool for understanding the basics of modern internal computer architecture and how peripheral devices are connected to this architecture. Many of these details are hidden on "quick-access" hobby platforms like the Arduino, or semi-single-purpose programmable ICs like WiFi interfaces. Also, the 8051 processor is the most popular 8-bit, inexpensive microcontroller, and it shows up in many products you may be excited about, e.g., video game controllers. The R31JP will show you what is really going on inside a modern computer. The board operates using 5 volts, developed with an on-board 7805 voltage regulator. The board includes random access (easily changed) RAM memory, ROM or read-only memory that holds the MINMON operating system, an LED light bank connected to port 1 (controlled by register P1), and connectors for RS232 serial communication and ribbon cables to connect to the briefcase kit. Do NOT remove IC's or components from the board, and do not alter the board, other than to connect a serial cable or ribbon connectors to the kit. The R31JP exposes an address and data bus architecture that is inherent but hidden from the user in many "all-in-one" processors available today. We will use the R31JP to learn about the hardware architectures commonly used to implement computing platforms. We will also use the fact that the R31JP can be connected to the briefcase kit to expand the system with new integra ted circuits on the kit breadboard using the address bus, data bus, and control lines available from this board. Be VERY careful to make sure that you've made reliable, good connections with the two ribbon cables (on connectors labeled K1 and K2, discussed more shortly) and the serial port. As with all our boards, there are exposed solder connections on the bottom of the board. Be careful where you set it down to make sure that you do not short connections on this board or any of our other microcontroller boards. **Do NOT break your R31JP!**

The **PSoC 5LP "Big Board,"** forged by Infineon/Cypress in the heart of a dying sun, is your most powerful and flexible piece of microcontroller hardware. The board is built around a Cypress PSoC 5LP, a powerful, state-of-the-art ARM core (Google it) processor embedded in a remarkable collection of reconfigurable hardware that can be programmed to yield different hardware functions. It offers the type of reconfigurability that would be desirable for making a liquid-metal Terminator. It MUST NOT BE BROKEN. Therefore: Do NOT hook this board up to anything "external" except for the USB programming cable to your PC, the serial Kable from your Kit that we'll discuss, and/or the blinky LED board, also to be discussed. You may make minor wiring on the board itself with great care. We will do this in some lab exercises; wait until we bring it up. We would prefer that you do no other wiring on the board. Beware: as configured, **this board operates internally on 3.3 volts**, not 5. This is no issue at all as long as you observe the restrictions above; you won't notice the board voltage. This board is your most likely target for a great final project, using the board as a "co-processor" connected to a PC. The board could play a game, or implement a simple MINMON-style assistant or "MATLAB" or file or signal processing system. **Do NOT break your Big Board!**

Finally, your **PSoC 5LP "Stick"** is the only board that you should really consider for connecting to anything "unusual" like a transistor driving a (small) motor. **We still recommend against this**. But, in principle, the "Stick" could be used on your breadboard, powered by a USB connection (from any USB power supply) if you have or can find a cable with a "female" USB end to connect to the programming fingers after the board has been programmed normally. When powered by a USB power supply, the "Stick" should be operating at 5 volts and can still be inserted into your breadboard if you leave the "USB fingers" hanging over the breadboard edge. We will use the Stick to create a Kovid Konsole for our early hardware experiments with PSoC.

Let's begin by taking a look at the R31JP board and the associated briefcase kit, henceforth just "kit" or "Blackbird," and how we connect the board and kit to build systems and experiment with the 8051.

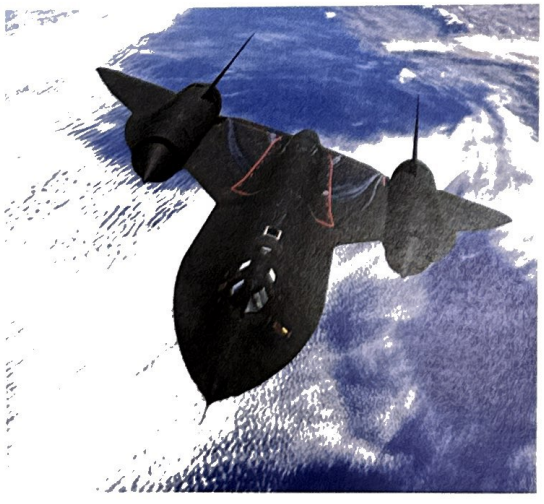
KIT CONNECTIONS:

Connect your R31JP board to the RS232 port of an IBM PC compatible personal computer using a DB-9 serial cable. If your computer has a serial port as in 38-600, you use a male-to-female DB9 serial cable to make the connection. Most likely, your personal computer or laptop does NOT have this serial port – which is no problem. We have included a USB-to-Serial adapter Kable in your kit, shown below:

Like the rainbow bridge Bifrost that connects Asgard to Midgard (Earth), the Kable (shown above) will permit you to connect your R31JP (or PSoC 5LP "Big Board," later) to your Windows 10 PC. Just as the ever-vigilant Heimdall guards the Bifrost, you too must impress Odin by configuring your PC, Kable, and microcontroller board to permit just and righteous communications to flow between your systems. We will look at the communication or "terminal" software more later, but for now, make sure you have the Kable and examine it. Note the USB type-A end that goes to your PC/Laptop, and the DB9 serial end that goes to your R31JP. NOTE: the USB type-A end operates at 5 volts. The DB9 serial end operates at a special cable voltage of plus and minus 9 volts. These ends, USB and DB9 serial, should ONLY be plugged into appropriate receiving connectors. If you are working on your own computer, the Kable for should work "fine" in Windows 10. It will also work in Windows 11, but may need a driver update.

During the term, you will use a 6.115 laboratory kit, *The Blackbird*, to complete laboratory assignments. The kit is a sophisticated instrument for electronics prototyping, tool storage, and experimentation with

your microcontroller board, the R31JP. The R31JP has been pre-installed on the kit and cabled with two 50-pin ribbon cables (one rainbow, one grey) to the function board on The Blackbird:

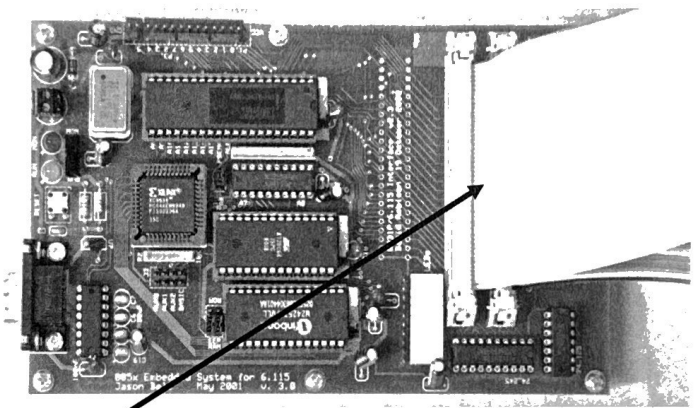


(Another famous Blackbird, but not your kit….)

The Blackbird is a delicate instrument, hand-assembled by the 6.115 staff to speed your ability to build and learn. We will NOT tolerate careless use of the kit that results in any mechanical or electrical damage to this expensive hardware. You are responsible for the care and proper use of this gear. ASK before you guess and do something that damages the kit. We are here and happy to help you.

To use the R31JP, power on your kit with the included power supply. Connect the power supply to the DIN connector, connect your R31JP board to the RS232 port of an IBM PC compatible personal computer using the DB-9 serial cable, and turn on the power switch at the top of the function board. Your R31JP board is pre-connected to your lab kit using the two 50 pin ribbon cables. Make ABSOLUTELY certain that the two 50 pin cables are connected, and do NOT remove them.
Pictorial reminder of correct connections:

Here's the R31JP board:

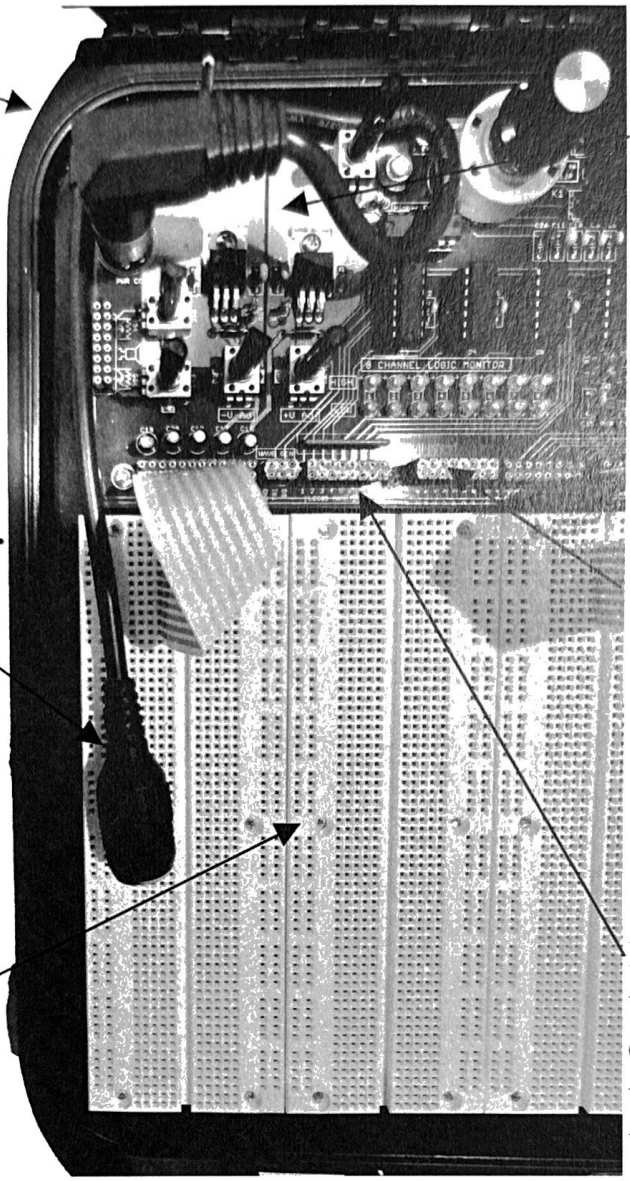


These ribbon cables are used to connect the R31JP board to your lab kit, as shown here.
It does not matter if you have one rainbow and one gray cable. Two gray or two rainbow cables will be fine. What is EXTREMELY important is that the R31JP is connected to the function board on the lab kit connectors in the correct order, i.e., the connector labeled "K1" on the kit to "K1" on the R31JP, and "K2" to "K2". This has been done for you when the kit was assembled. Do not alter the cabling.

The "function board" on the kit that is connected to the R31JP is shown in the figure below.
The function board on your Blackbird kit shown below has a silk-screen text listing printed right on the board. This list tells you exactly what each pin on the board corresponds to on the R31JP board, e.g., address lines A0-A15, data lines, etc. Please make certain that you understand what's going on with the kit connection, i.e., ask questions during the laboratory familiarization, so that you do not carelessly damage the R31JP or the lab kit! Other connections on the function board provide power for your breadboard, connections to the 5 volt TTL logic lights ("Logic Monitor"), signals from the on-board function generator, and other functions you can see by examining the board. Do NOT guess. Ask for help if you are confused about any of the connections on this board. Note, as shown below, that the breadboard rails are "split" in the middle, and need wire jumpers if you want the vertical rails of the breadboards to be continuous. Also, note that the double hole connectors at the edge of the function board are like normal breadboard holes. Do NOT USE OR FORCE larger wire here. Do NOT break or remove these connectors. See the staff immediately if you have any issue with the connectors.

Do not undo the DIN cable connector.

Do connect your power supply to the DIN end here.



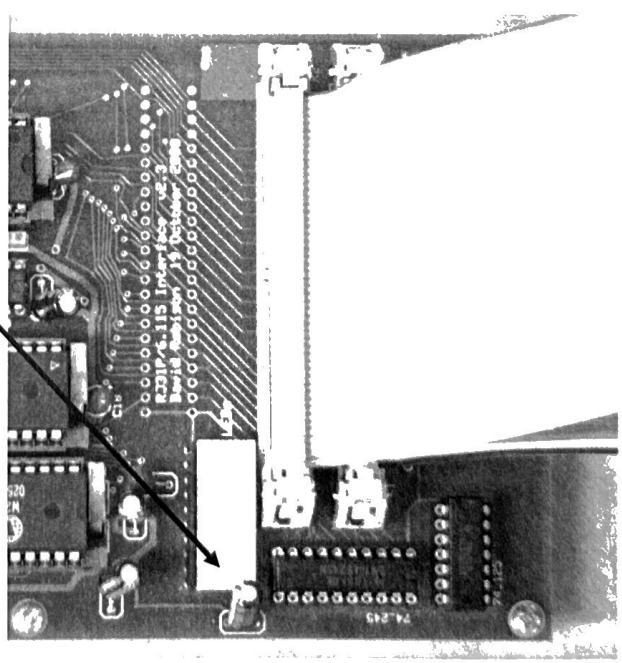
Do NOT solder on the function board, or remove ICs from the function board!

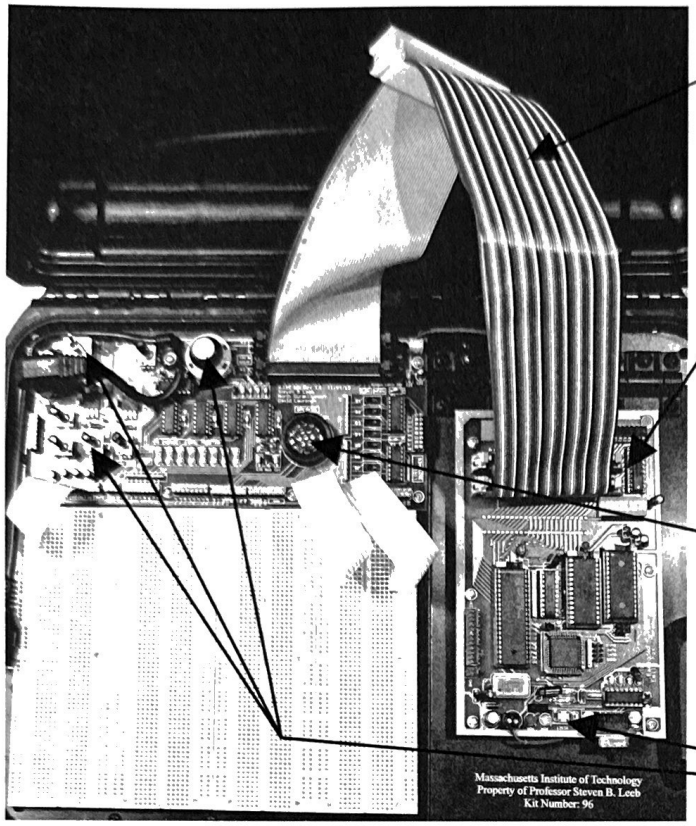
Only regular 22 gage hookup wire in these connectors!

The breadboard rails are "split" in the middle. Install jumper wires, e.g., here, if you want the vertical rail to be continuous from the top of the breadboard to the bottom!

Do NOT damage the double connectors on the function board, they are delicate!

Here's a close up of your R31JP microcontroller board connected to a kit. The LED light bank on this board contains 10 LED segments. Eight of these are connected to Port 1 on the microcontroller. The other two are not used.





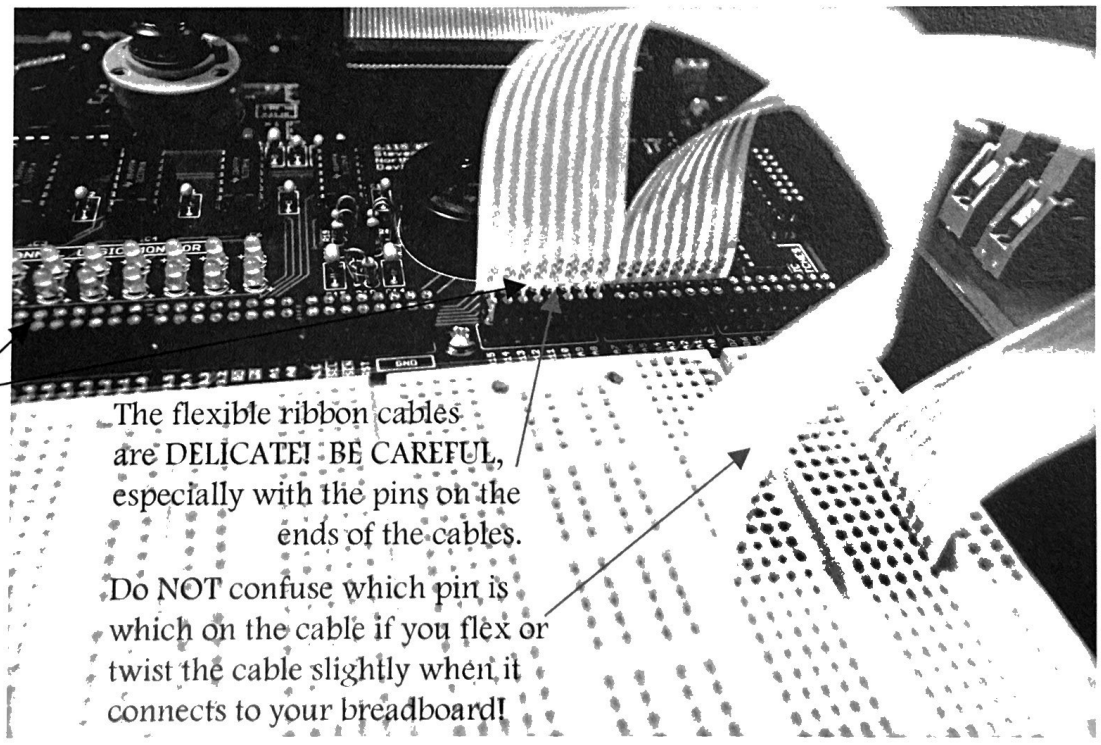
Do NOT remove, detach, or undo these ribbon cables.

Do NOT remove or damage your R31JP and do NOT disconnect the power cable to the R31JP.

The speaker on the function board is delicate - do not damage it mechanically.

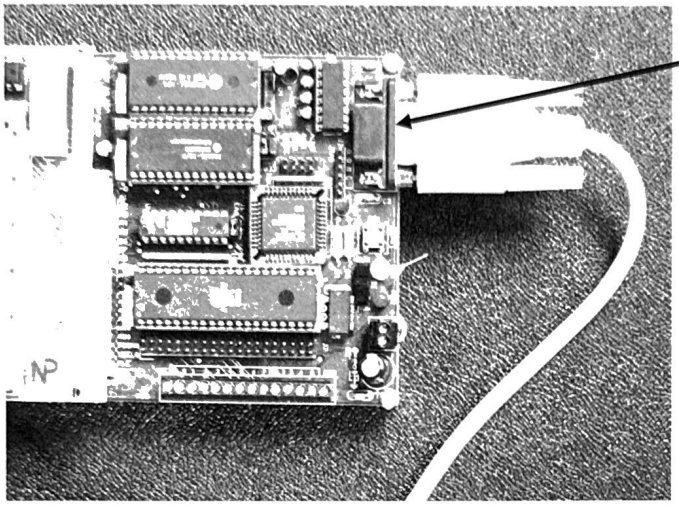
Be gentle with switches and mechanical connections!

USE ONLY 5 volt TTL logic level signals as inputs to the function board connectors and the R31JP!



The flexible ribbon cables are DELICATE! BE CAREFUL, especially with the pins on the ends of the cables.

Do NOT confuse which pin is which on the cable if you flex or twist the cable slightly when it connects to your breadboard!



The "DB9" connector on the R31JP board is used to connect the microcontroller to the serial port (typically COM1 or COM2) of a personal computer. The PC can be used to compile programs, download them to the R31JP, and see results from the microcontroller.

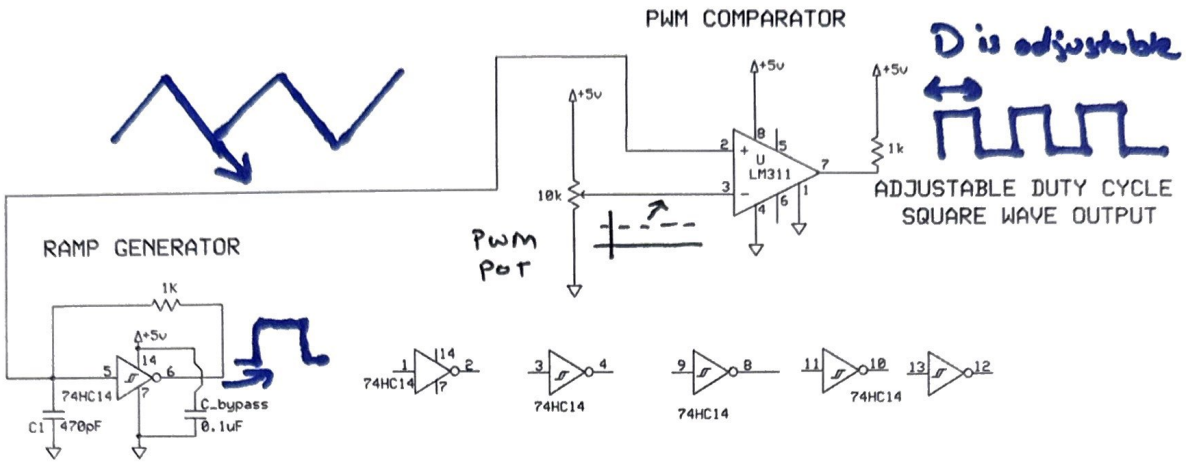
There are two essential features of your kit that you should know in order to avoid frustration. First, there are several DC power supplies on the function board. A fixed five volt supply and two variable power supplies (0 to 12 volts and 0 to –12 volts) are available at the top of the kit. There are also two more fixed power supplies for plus and minus 12 volts. Use the five volt power supply to power chips on your breadboard, and use the plus and minus 12 volts to power analog chips like op-amps. The R31JP is automatically powered from the five volt supply on the function board, do not interfere with this connection to the R31JP or alter the R31JP power supply from the function board. The breadboard rails are NOT automatically connected to the power supplies. You need to connect these rails to the top power supplies and ground as you see fit. USE BYPASS CAPACITORS to filter your power rails on the breadboards.

BUILDING ON THE KIT:

Designing an electronic circuit "from scratch" is a different mental exercise than analyzing a completed circuit that is given to you. To **design** a circuit, you need, among other skills, to be able to quickly analyze the range of effects that can be created by changes in component values. You need to find the right circuit analysis technique that allows the quickest evaluation of trade-offs in any given design problem. Certain analysis techniques are better suited for use under certain conditions than others. A wise choice of analysis technique for a proposed circuit topology can make it quick and fun to determine the capability or performance of a circuit topology under different conditions.

Perhaps most important: what you think you designed is not necessarily what you built. High performance circuits – circuits operating at high frequencies or that try to make subtle measurements of small signals or that convert between analog and digital worlds at high speeds – have an insidious way of exposing the "parasitic" behaviors we usually try to ignore, e.g., the inductance of a wire, the resistance of a capacitor, the resistance of a connector, and so on. It is a skill and an art to make the "built" circuit actually behave according to the electrical design intent.

Let's "take the kit for a spin" and see what happens when I try to build a circuit on the breadboard. This circuit does NOT involve the microcontroller yet, and uses just a few parts. (**You don't have to build this,** just "watch me," please, for now.) The circuit is shown in the annotated schematic below. It's not critical that you understand every detail of the circuit right now, i.e., don't panic if you don't recognize these chips. It's just an example we will use to discuss "proper usage" of the kit:



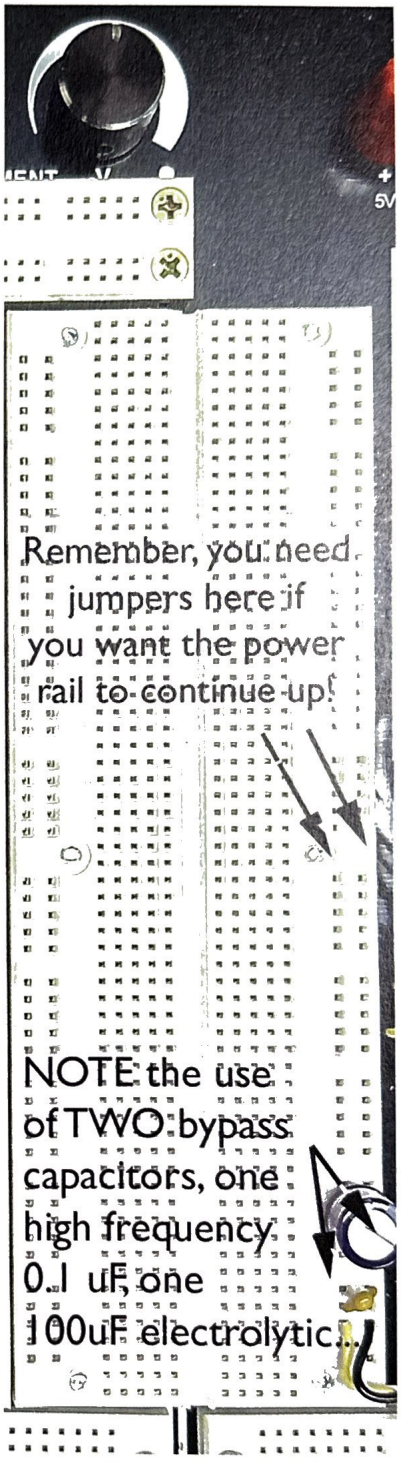
In overview, the purpose of this circuit is to create an output square wave with an adjustable duty cycle. The duty cycle is the fraction (between 0 and 1) that the output signal is high. The circuit provides this function by using one 74HC14 gate (there are six total on the integrated circuit) to make a fixed duty cycle square wave (on pin 6) and an associated triangular-style wave (on pin 5). The triangle wave from pin 5 is fed to one input of an LM311 comparator. The other input of the LM311 comparator is connected to a DC level created by a variable resistor or POT. When the triangle wave is "above" the level of the DC level, the output of the LM311 will be HIGH. When the triangle wave is "below" the DC level, the output of the LM311 will be LOW. By adjusting the POT to move the DC level, we can change the duty cycle of the output square wave on pin 7 of the LM311. (Pretty cool!)

Note the proposed use of a **"Bypass Capacitor"** in the picture above. Bypass capacitors are important to place between power and ground near any components that draw "surges" of current for switching events. The bypass capacitor helps keep a voltage rail that is supposed to be "flat and unchanging" stay "flat and unchanging." The bypass capacitor acts as a local reservoir of charge to supply transient demands quickly. The exact value of the bypass capacitor, typically around 0.1 uF or so, is not as important as ensuring that the bypass capacitor is small (for low parasitic inductance) and a quality, high frequency part (metal film). It is common to implement bypassing with a parallel combination of electrolytic (good charge storage) and metal film (good high frequency performance) capacitors.

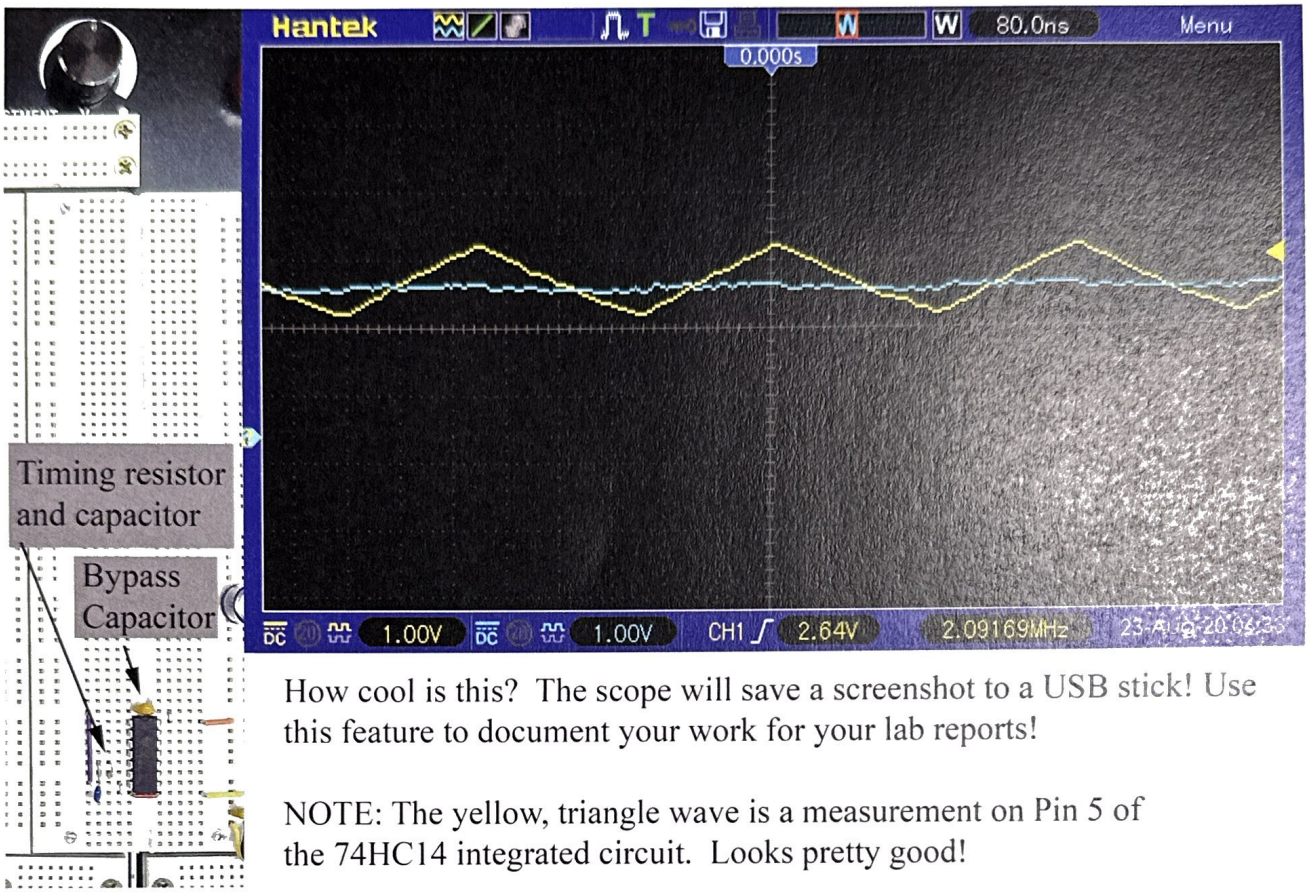
So you can see how I started wiring my kit in the picture on the right! First, I spun a piece of yellow and black wire together to make a twisted pair. The twisted pair minimizes loop area and keeps stray inductance and magnetic pickup low. One end pair (yellow and black) goes to the kit +5 and ground (not shown). The other end pair (yellow and back) energizes a kit breadboard rail. Note that I am only using the bottom half of the breadboard. I would have to add rail jumpers to continue the rail all the way up.

Note also that I added bypass capacitors (one electrolytic, one metal film) immediately on the rails where I connected the output of the common mode chokes.

These capacitors are "nice" but they may not be sufficient. Good bypassing should involve placing capacitors very close to each device that is switching or drawing "slugs" of current.



Next, I wired up my 74HC14 according to the proposed schematic. I wired WITH THE POWER OFF. But, when I finished the 74HC14 wiring, I decided to check the circuit with my oscilloscope. It's always a good idea to work "modularly," checking subsystems as you go. It's harder to debug a huge system that you just built, all at once, and suddenly turn on. Here's what I did and saw:

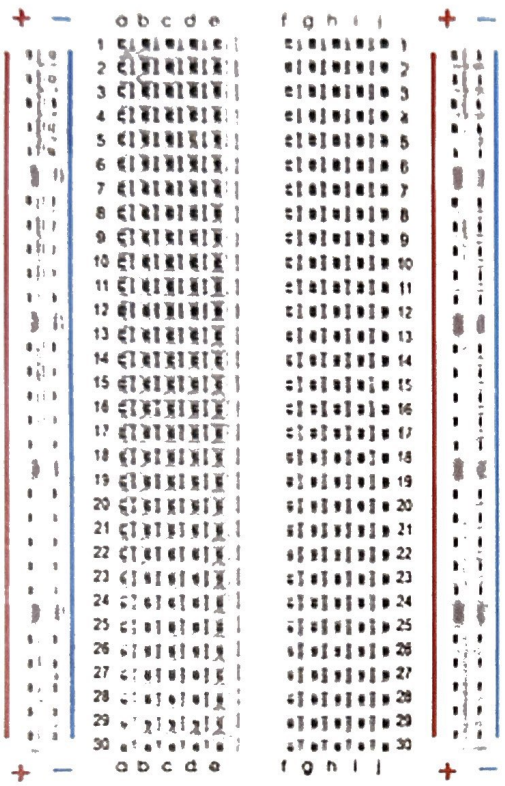


How cool is this? The scope will save a screenshot to a USB stick! Use this feature to document your work for your lab reports!

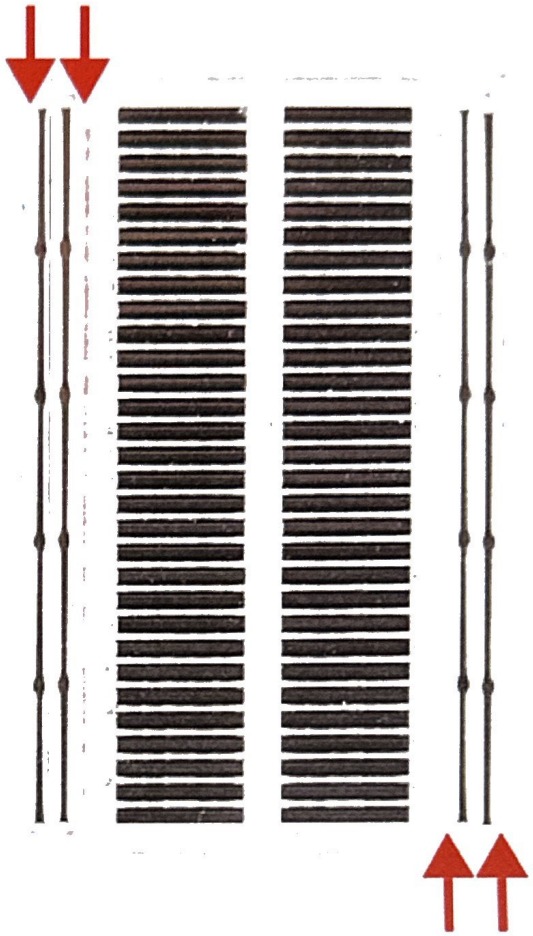
NOTE: The yellow, triangle wave is a measurement on Pin 5 of the 74HC14 integrated circuit. Looks pretty good!

So far so good! The triangle wave on the 74HC14 looks ok (yellow trace on scope Ch1). I'm nervous, however, because I remember and see that my design is attempting to create a (roughly) 2 MHz waveform! That's too fast for our construction technology, for the most part. It's also approaching fast for our oscilloscopes, which have a bandwidth of 100 MHz. So, while I trust the measurements, I'd be worried about the oscilloscope if we were going much faster, and I am definitely concerned about the build. Remember, the breadboard consists of row after row of metal sockets. Each row has a capacitance (5 to 10 pF) with the neighboring row. Fast waveforms spread signals all over the breadboard through these parasitic capacitances

You can see a photograph that visually explains the issue with the breadboard on the next page:



Front side of a breadboard....



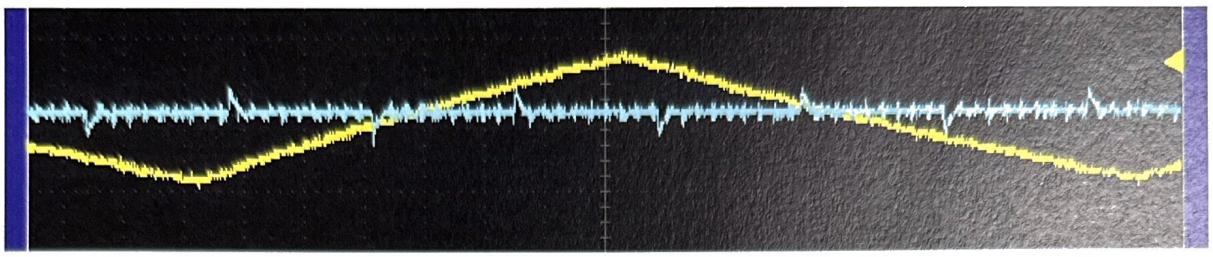
.... and the back side after concealing paper is removed.

The breadboard "rows" and "power rails" are implemented with metal clips that you can see on the "back side" of the breadboard (after we removed a concealing paper). The clips make the "connections" between the relevant holes on the breadboard, which of course makes the breadboard convenient to use. However, the metal "clips" basically function as capacitor plates, creating a web of parasitic capacitive connections between every node to every other node on your breadboard circuit. This can become an enormous problem at high frequencies.

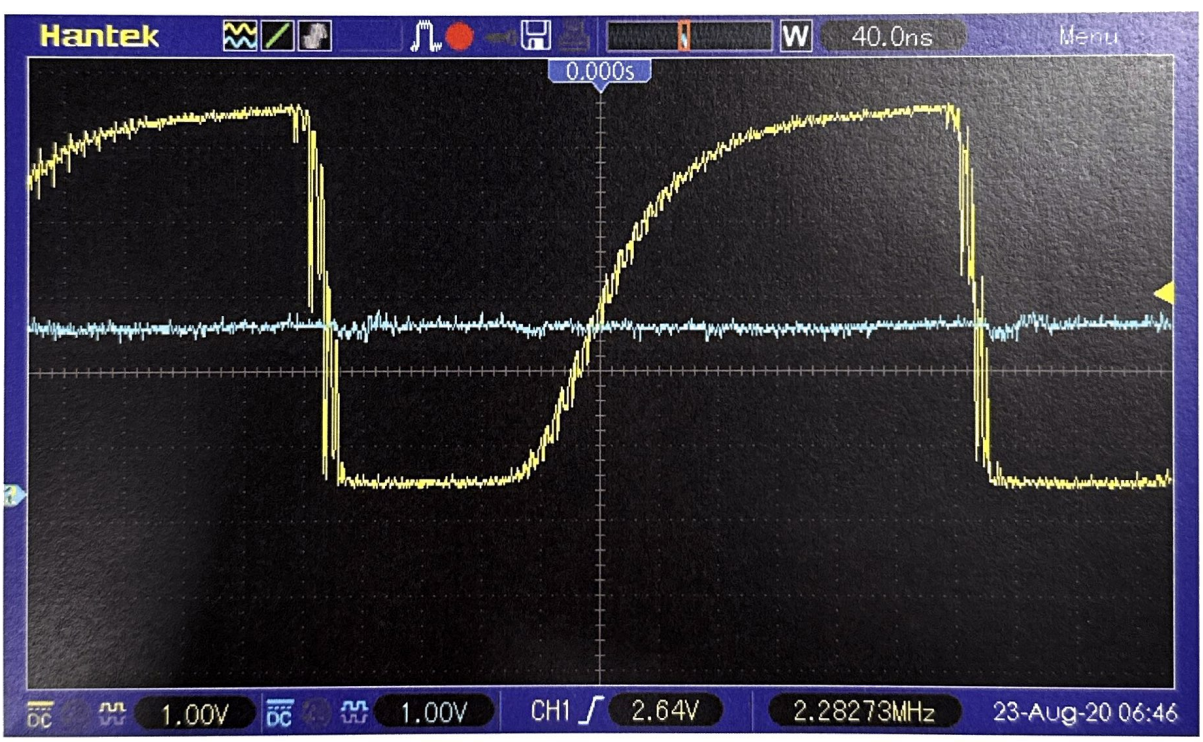
But I decided to press ahead rather than re-evaluate my design, in order to see what might happen. So my next moves were to wire up the "PWM POT" and the LM311 comparator.

The blue trace (Ch2) in the scope photo above shows the "DC level" from my PWM POT. The LM311 is supposed to compare the yellow (triangle) trace with the blue (level) trace and create an output square wave with a variable duty cycle, where the duty cycle is controlled by the POT. What do you suppose happened when I finished adding the components? Results are on the next page:

This next scope photo shows the triangle and level waveforms on pins 2 and 3 of the LM311:



I see trouble. The blue line is "fuzzy", it's contaminated with switching events and cross-talk signals couple by parasitic breadboard capacitance. The LM311 will be confused. Remember, the LM311 compares the yellow and the blue traces. It's trying to give a clean output square wave when one waveform is solidly above or below the other. With all the "fuzz", as the yellow triangle crosses the blue level, the comparator offers a confused response, repeatedly triggering around the messy comparison. Here's what I see on my kit with the scope watching the output pin 7 of the LM311 on Ch1 (yellow):

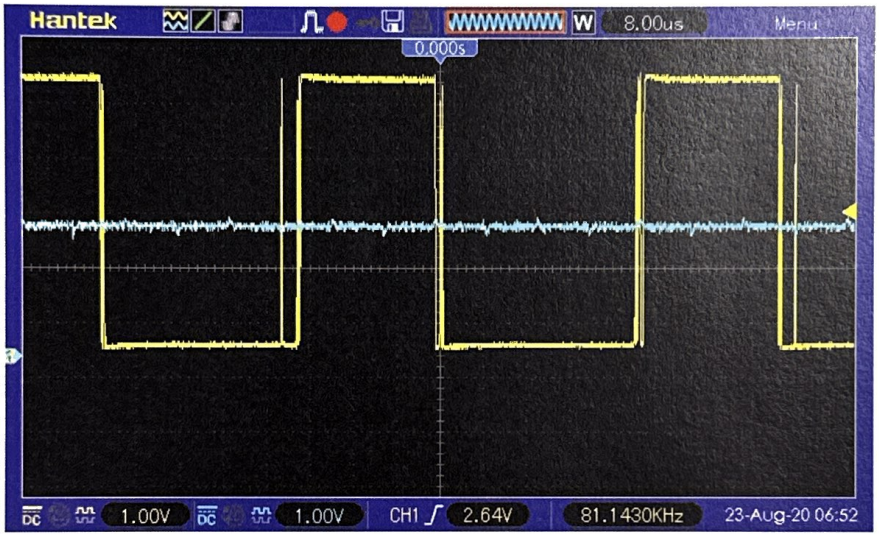


There are at least two big problems. First, as best as I can trust the scope at these frequencies, the rise time of the "square wave" is anemic. The LM311 may be able to run at 2 MHz (check the datasheet), but the parasitic capacitances on the breadboard slow everything down, creating a gradual rise instead of a sharp edge. That may or may not be a problem depending on how I plan to use the wave later.

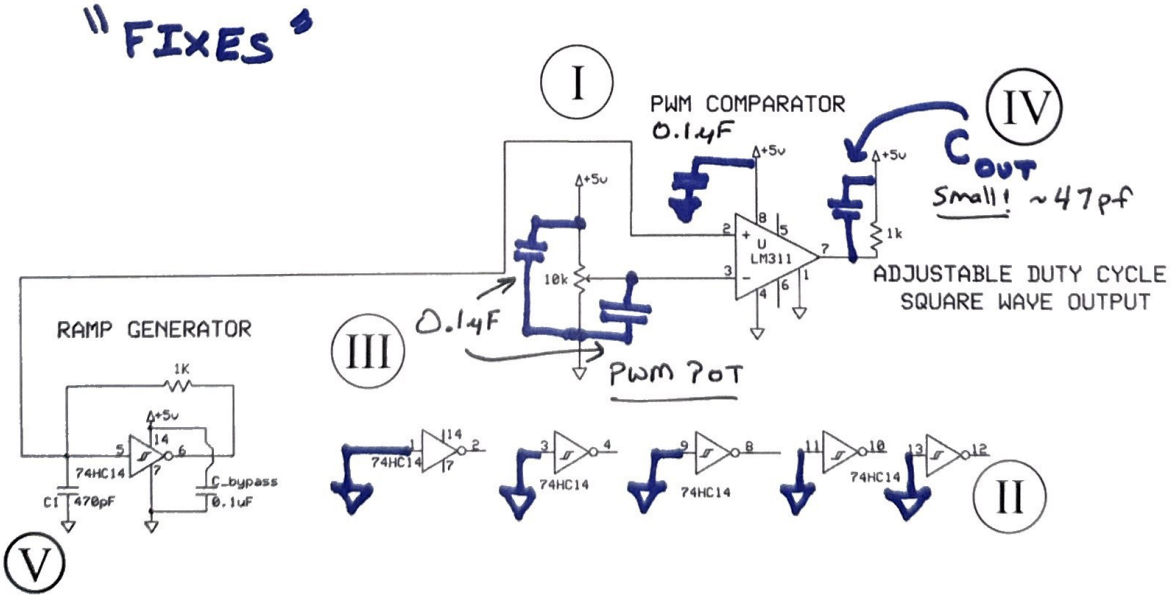
Second, and perhaps more worrisome, is the repeated triggering or jumps, particularly on the "falling edge" of the square wave. The wave is not crisply defined. If we used it to drive a power transistor or a microcontroller input, the transistor might try to turn on and off very quickly in a short time, somewhat like flicking a light switch on and off very quickly. Many devices might be injured by a sloppy control

input like this. A microcontroller program could easily become confused looking at a waveform like this, thinking that the waveform changed state multiple times when we only intended for one state change.

I tried to address both of these problems on my kit. First, I imagined that I could change my design to run at a lower overall frequency. If this were not the case, I might have to give up on breadboard construction and move to a printed circuit board. But let's assume here that I could be much less aggressive with frequency and still be happy. To lower the frequency, I changed the capacitor on Pin 5 of the 74HC14 to a larger component, slowing down the overall oscillation frequency to a much lower value around 80 kHz. Here's what the LM311 pin 7 output looks like in this case:



At the lower frequencies, the rise and fall times look comparatively crisp, and I have what might serve as a better approximation of a square wave. Unfortunately, however, I still have the problem with "fuzz" on the input waveforms of the LM311. The comparator is indecisive near the edges, and I get glitches and multiple transitions on both the rising and falling edges of the square wave. Not good. To clean up this problem, I have to be much more disciplined with my build to control all of the possible problems that might be causing "fuzz" in the waveforms and ruining the LM311 output:



A summary of the fixes:

I. First (Roman numeral I in the annotated schematic above), I added bypass capacitors to make sure that every IC (including the LM311, and all along the power rail) were properly bypassed.

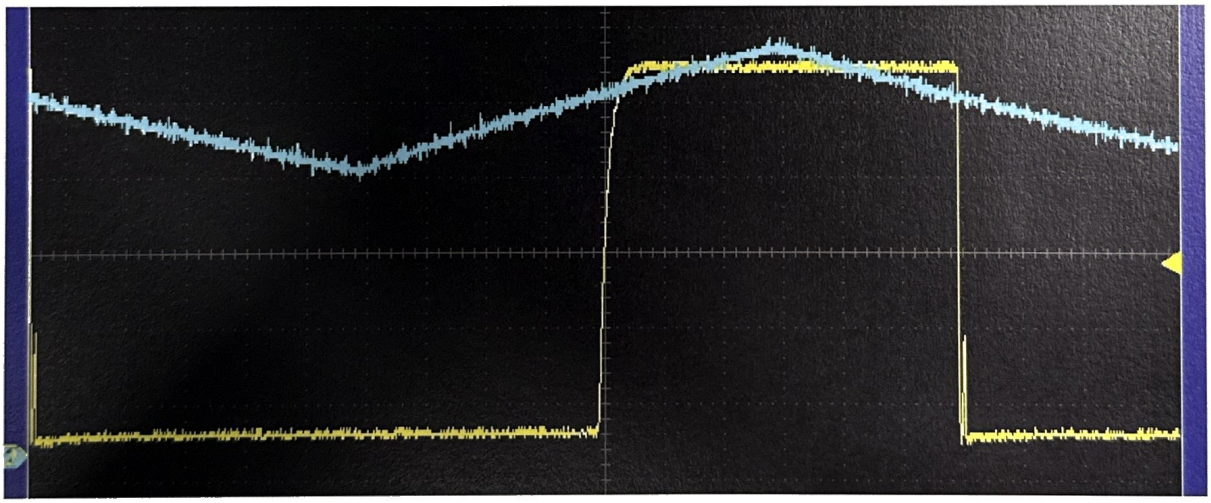
II. I grounded any "unused" inputs to logic gates. NEVER leave a gate input floating. You don't necessarily have to connect an output to anything, but ALL inputs must be connected to something, e.g., another gate, ground, appropriate logic "high" voltage rail, etc. If you leave an input unconnected or "floating", it may cause the gate to oscillate wildly in response to ambient charge and coupled signals.

III. I also added bypass capacitors around the POT to keep the DC level as flat as possible. (If the input had been an audio signal, or something other than DC, I might not have been able to add these capacitors without introducing unwanted filtering.)

IV. The LM311 output glitches or oscillations can be reduced or eliminated by adding "positive feedback" around the comparator (read the datasheet). I chose a "quick and dirty" trick of adding a small output capacitance in parallel with the output pull-up resistor connected to pin 7. This adds a capacitive load to the LM311, a relatively inelegant solution compared to the positive feedback approach. But it works.

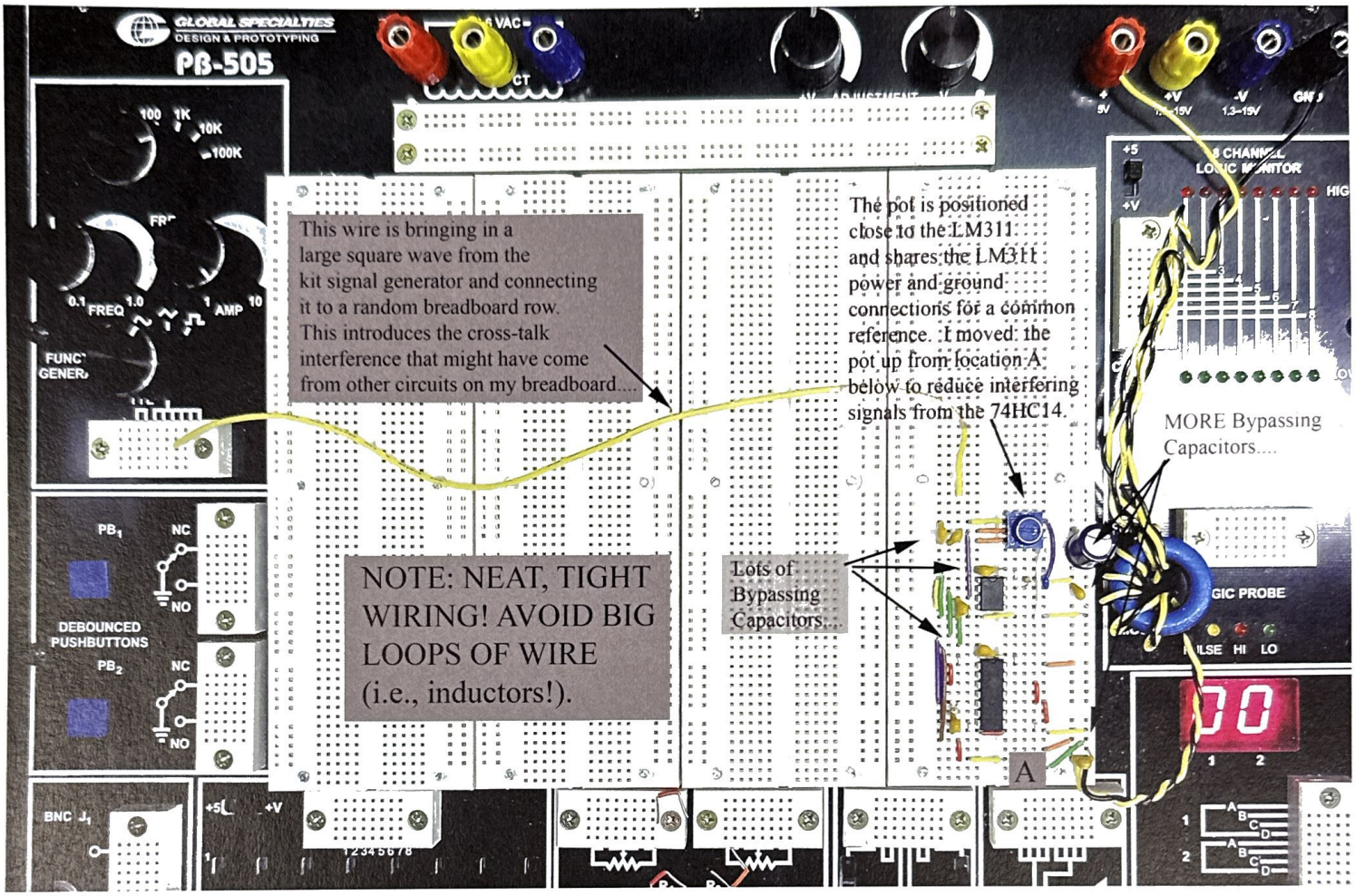
V. And, as we discussed above, I slowed down the overall design by increasing the capacitance at pin 5 of the 74HC14.

All said and done, I wind up with an LM311 output waveform that looks like this:



The "blue" Ch2 trace shows the triangle wave going into pin 2 of the LM311, and the "yellow" Ch1 trace shows the output on pin 7 of the LM311. The rise and fall times are adequate for many of our uses, and the glitches have been sufficiently reduced that this waveform is useful. Note that we will often follow the output of an LM311 with more logic circuitry (e.g., the "delay and delay_bar" circuits we discuss in this lab) that adds a natural de-glitching or low-pass quality that helps block issues in the LM311 output.

A picture of my "final" kit assembly is shown on the next page. **NOTE**: this kit is a little different than the layout on your Blackbird, but the essential points are the same. The "random wire" coming from the signal generator is intended to model the effect of other circuitry that might have been present on a busier kit. The signal generator is driving a randomly chosen breadboard row, and introduces a signal that can create cross-coupling; this is a good test or "challenge" to make sure that the "fixes" work well.



Take some time to familiarize yourself with your kit. Note all the cool features available (the Blackbird offers signal generator, switches, etc.) and keep them in mind for when they might be of use to you. Treat the kit carefully. Your parts also include spare slices of breadboard in case you get pressed for space.

NOTE: As we work through our exercises this term, you may think you need a component value that you do not have in your kit. Generally, you can get parts in 38-600 and 38-500. But, if you are stuck in a dorm room, independence and intrepidity, please! For example, you may be able to "make" a 5K resistor out of a 1K in series with a 3.9K. Think about where values in your designs have to be "exact" or "adjusted" to give needed performance (e.g., timing clocks) versus where something "close" is good enough (e.g., bypass capacitors: 0.1uF, 0.022 uF, 0.22 uF – likely "all good"). Understanding the intent of what you are building will be essential.

DO NOT remove circuits that you build for a lab until you have checked off the lab. That is, keep everything you build on your kit so it's available to show us at checkoff. Work neatly and plan ahead. We will reuse some circuits in later labs, so it is to your advantage to work neatly and keep useful circuits on your kit, e.g., the keypad reader you will build in Lab 1.

BYPASS YOUR CIRCUITS (modified from Wikipedia):

One more time, because it's important:

A **decoupling capacitor** is a capacitor used to decouple one part of an electrical network (circuit) from another. Noise caused by other circuit elements is shunted through the capacitor, reducing the effect they have on the rest of the circuit.

For example, if the voltage level for a device is fixed, changing power demands are manifested as changing current demand. The power supply must accommodate these variations in current draw with as little change as possible in the power supply voltage. When the current draw in a device changes, the power supply cannot respond to that change instantaneously. As a consequence, the voltage at the device changes for a brief period before the power supply responds. The voltage regulator adjusts the amount of current it is supplying to keep the output voltage constant but can only effectively maintain the output voltage for events at frequencies from DC to a few hundred kHz, depending on the regulator (some are effective at regulating in the low MHz). For transient events that occur at frequencies above this range, there is a time lag before the voltage regulator responds to the new current demand level.

This is where the decoupling capacitor comes in. The decoupling capacitor works as the device's local energy storage. The capacitor cannot provide DC power because it stores only a small amount of energy but this energy can respond very quickly to changing current demands. The capacitors effectively maintain power-supply voltage at frequencies from hundreds of kHz to hundreds of MHz (in the milliseconds to nanoseconds range). Decoupling capacitors are not useful for events occurring above or below this range.

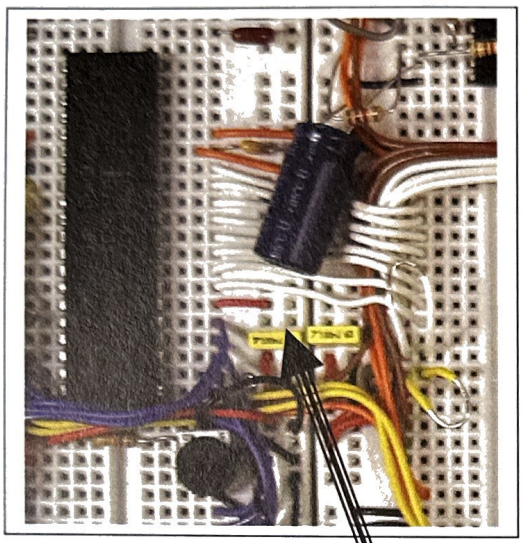
An alternative name is **bypass capacitor** as it is used to bypass the power supply or other high impedance component of a circuit.

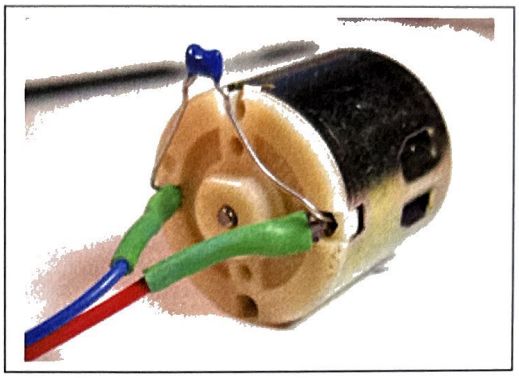
Essentially: Bypass capacitors are local "buckets" or "batteries" of charge that smooth out the power for your IC's. They must be used in power electronic and digital circuits, and are generally a good idea. They provide local "hold up" of the voltage for each IC, and limit the spread of transient voltage disturbances around your circuit.

Key points:

- **Use a small metal film or ceramic cap for bypassing right near each chip, e.g. connected directly between the power and ground pins of each chip** (e.g., on a 14 pin 74XX logic part, connect a bypass cap between pin 14 and pin 7).
- The value of the capacitor is not nearly as important as its physical size. It should be a small capacitor with good high frequency performance typically in the range of 0.01 to 0.1 uF.
- A mix of caps, e.g., an electrolytic in parallel with a metal film or tantalum capacitor can be used to provide good high-frequency bypassing plus good DC hold-up at low frequencies and for longer transients.
- Be CERTAIN to OBSERVE POLARITY AND VOLTAGE RATINGS!
- READ about BYPASS Capacitors in Scherz (find the topic by using the Scherz book index).

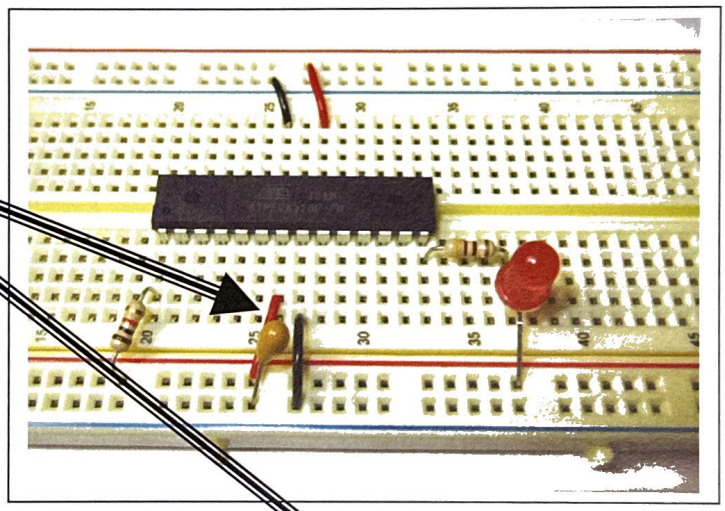
Pictures of adequately bypassed IC's and a motor:

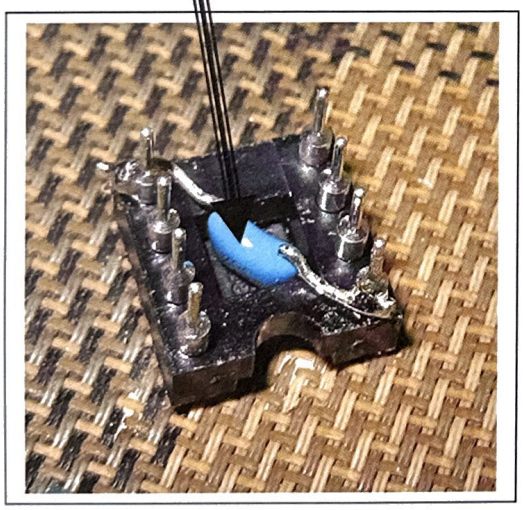


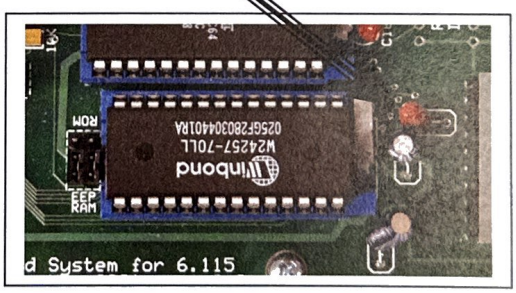


BYPASS CAPACITORS positioned between IC power and ground...

Bypass "built in" to an IC socket:







TERMINAL COMMUNICATION SOFTWARE AND KABLE:

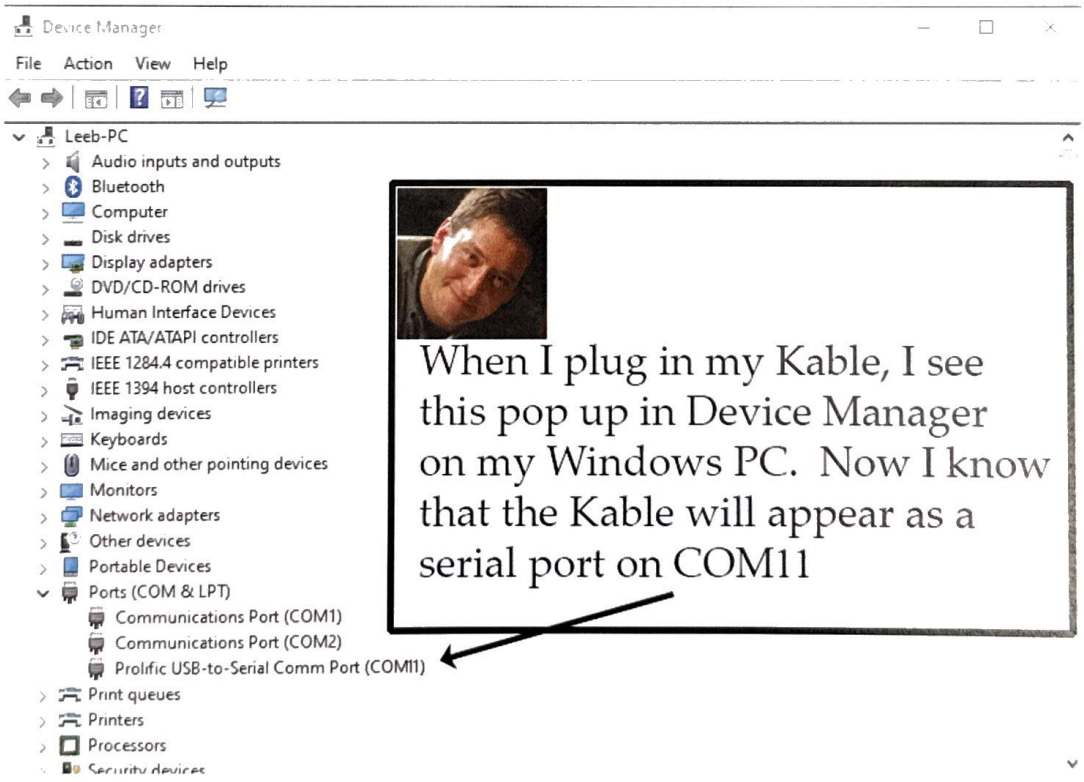
So, how do we get a Windows 10 PC to talk to one of our microcontrollers boards? Let's think about the R31JP. We will use the Kable to connect the R31JP serial DB9 port to a PC-side USB port. But how do we actually "talk" to the R31JP from the PC? For example, how would we download a HEX file made by AS31 to the R31JP? Easy! We will use a terminal program! There are many terminal programs that work equally well – generally, when a lab document referes to "Hyperterminal" or "WinSCP" or "Putty" or "TeraTerm" or "RealTerm" and so forth, we mean "a terminal program" – they all provide similar function for our purposes.

There are a lot of choices for terminal programs, pick one that works for you. We have posted three free programs at:

web.mit.edu/cdev

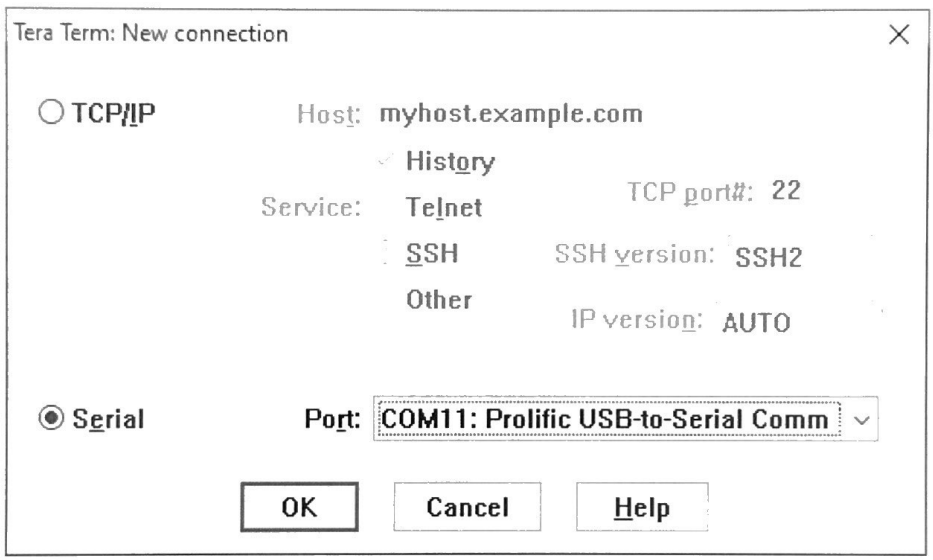
If you check the website, you will find "Putty," another program called "TeraTerm", and a program called "Realterm". They are all excellent, with different features. We will recommend TeraTerm and Putty for most of our work. Realterm is more complex and full featured, so you should keep it in mind if you find special needs in your final project. Install TeraTerm and Putty on your Windows 10 PC.

On the staff Windows 10 PC (Uncle Steve's desktop computer), the Kable is "plug and play". Simply inserting it into a free USB port resulted in the Kable appearing as a serial "COM" port in the Windows Device Manager. If you need to, you can use the Kable package CD to install drivers, but we have not found this to be necessary. When I plug the Kable into a USB port on my Windows 10 PC, here's what I see in Windows "Device Manager" under "Ports (COM & LPT)":

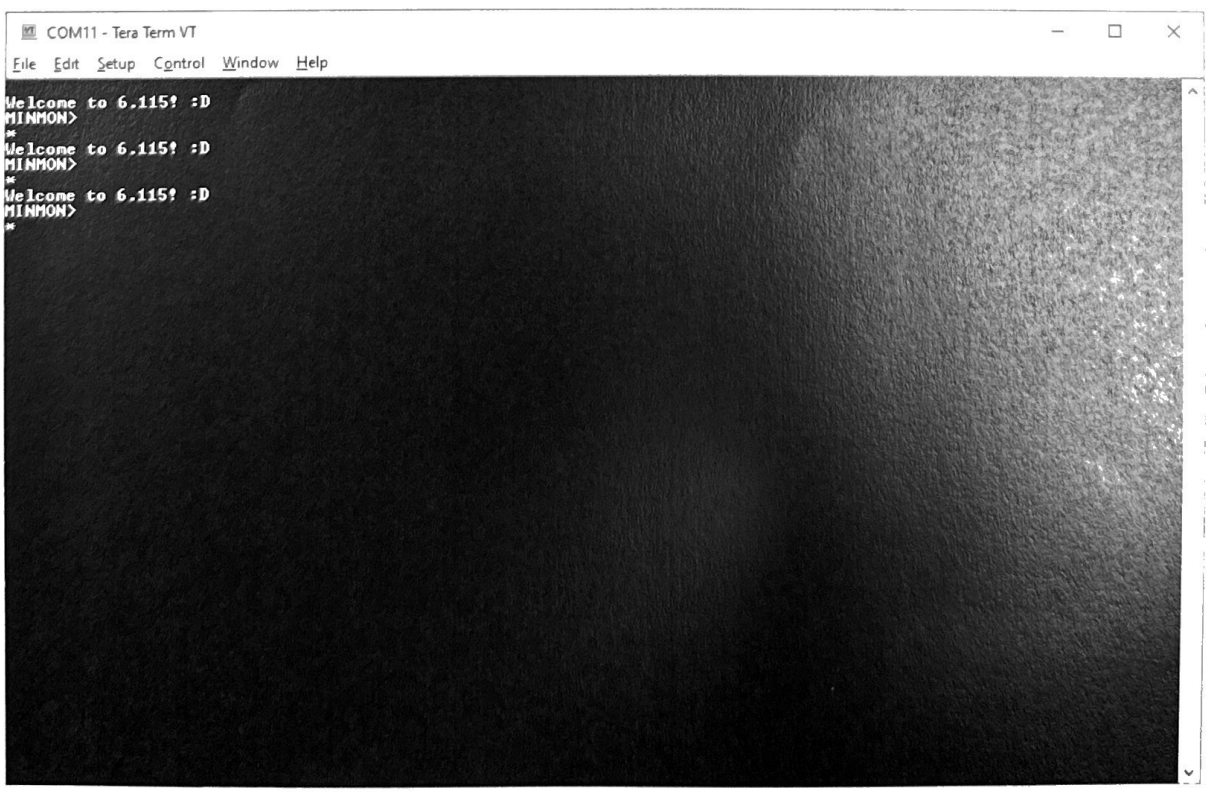


Now, use a "terminal" program on your Windows 10 PC to configure a terminal window to talk on the COM port (COM11 above, but this will vary) at 9600 baud, 8 data bits, 1 stop bit, "none" for parity, and "none" for flow control.

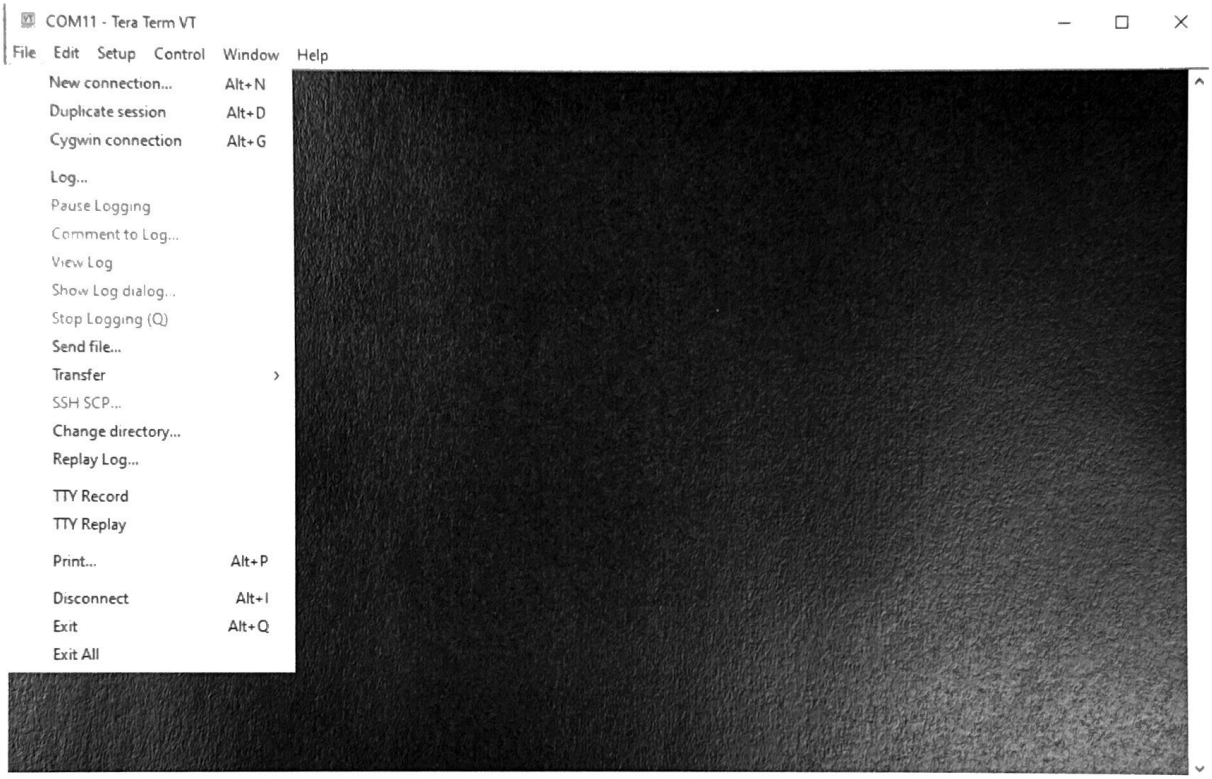
Here's an example interaction using TeraTerm. When you start the program, you'll be greeted with a "New Connection" window like the one shown below. Assuming your Kable is plugged into a USB port on your PC, choose "Serial" in the window, and use the "Port:" pull-down menu to select the COM port associated with your Kable. (Be sure you choose the right COM port for your Kable in the pull-down menu.)



Once I've selected "OK" with the correct port, I will be given a terminal window that I can use to talk to my R31JP when it is powered up. A typical interaction looks like this:



I pressed the white "reset" button on the R31JP three times in the example above. With the MON/RUN switch set to "MON," the R31JP runs our mini-operating system, called MINMON (mini-monitor). We will talk more about MINMON in class. For now, be aware that MINMON offers us a "prompt" in the TeraTerm window, and I see the "greeting" ("Welcome to 6.115!...") from the R31JP three times, once for each of the reset presses, in the window above. I could also have used Putty instead of TeraTerm to "talk" to the R31JP. Note that TeraTerm has several useful functions. For example, under the "File" menu, TeraTerm offers the option to "Send file...." :



I could use this option in order to "transmit" a HEX file from the PC over to the R31JP. Press "D" in the TeraTerm window to tell the R31JP that you want to start to ("D") download a program. Then use the TeraTerm "Send File" option to transmit a HEX file to the R31JP. We will see a demonstration during the familiarization. Run the HEX file program you just downloaded by using the command "G8000" in the TeraTerm window, or by switching the R31JP from MON to RUN (small switch on the R31JP board) while holding down the white reset button on the R31JP.

Beware: If you plug in other USB devices, it is possible that you're serial COM designation for Kable will change, and the TeraTerm communication will appear to have suddenly stopped working. If this happens, check Device Manager again and restart TeraTerm with the correct COM port.

Also: TeraTerm naturally defaults to serial communication with 9600 Baud, 8 bits, no parity, 1 stop bit, and no flow control; this is exactly what we want to talk to the R31JP. We will talk more about these settings in class. But, if you ever need to change the settings, you can do so in TeraTerm by accessing the "Serial port setup and connection" menu from the menus under: "Setup → Serial Port...".