LAB 5
Timing and Countng

Suggested Reading

1. Using the RCTime command

The PIC micro-controller has a built-in 16-bit timer. the PBASIC language accesses this timer through the RCTime command.

```
RCTIME pin#, state, variable
```

This command measures the time it takes a pin to change from high to low, or from low to high and puts this value in a variable.

For example

```
RCTIME 1, 0, t
```

would measure high long it takes pin 1 to change from the low state, below 1.4 V to a hight state, above 1.4 V. After execution of this command, the value in t is the time, in 2 μs intervals for the basic stamp 2. (Other models of the stamp have different time intervals.) If the pin is already in state 1 when it is executed, then the value put in t is 1. The largest time is 65535 intervals (131070 μs for Stamp 2) and if the pin remains 0 for longer than the longest time, then the value in t is zero.

The following circuit shows how this could be useful for measuring resistance or capacitance.

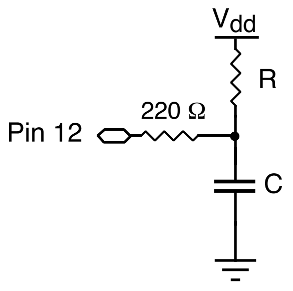

Fig 1: This circuit can be used with the RCtime command to determine the time it takes the capacitor to charge from 0 to 1.4 V.

The following commands are used to active it

```
LOW 12
PAUSE 100
RCTIME 12, 0, t
```

The LOW 12 puts pin 12 in the low state and thus discharges the capacitor. The DELAY command should give enough time for the capacitor to discharge through the 220 Ω resistor. (You should check this.)
The capacitor charges from 0 V through the resistor and the voltage increases in the following way:

$$V(t) = V_o[1-e^{-t/RC}]$$

V_o will be 5 V. So you can easily show that the time for the voltage to increase from 0 V to 1.4 V is 0.33

RC.

> Construct the circuit using a 1 kΩ resistor and a 0.1 μF capacitor. The predicted time constant is 100 μs. Measure the actual time returned by the RCTIME command and try to explain if there is any discrepancy from the expected value. How does the returned value compare with the theoretical with a 10 kΩ resistor instead?

You can see that if you have a capacitor of known value then this circuit could be used to measure unknown resistance values. Conversely if the resistor is known, then this could be used as a capacitance meter.

> Use the circuit as a resistance meter by replacing the 1 kΩ resistor with resistors of various values. Change the program to print the resistance in ohms or kilo-ohms. Check it's functioning for at least three "unknown" resistors comparing your results the nominal values and values measured by a bench-top ohmmeter. Discuss the limits (largest and smallest resistances for which it is practical) of the micro-controller ohmmeter and its accuracy at various ranges.

> Put a 10 kΩ resistor back in and try using the circuit as a capacitance meter in the same manner. Measure the value of several capacitors and determine the capacitance range over which it is usable.

2. Measuring light levels with a CDS photoresistor (or photocell) as light sensor.

A simple, inexpensive light sensor consists of a device made from CdS whose resistance and conductance depend on the light levels.

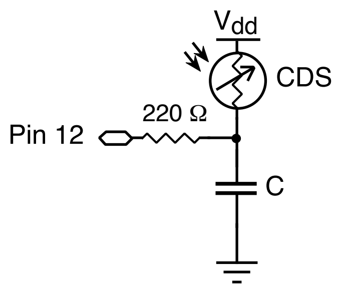

Fig 2: Replacing a resistor with a CdS photocell allows the RCTime command to measure light levels.

Measure the resistance of the of the CDS photo-resistor under a wide range of light conditions inside and outside the lab. Choose a value of C which will enable you to log the light level.

3. Counting and detecting physical events.

Light sensors are often used to detect physical events. For example a light source and sensor are placed on opposite sides of a doorway, When humans or other animals pass through, the decreased light level signals the passage of the beast through the doorway. One problem encountered with this technique is that when the beast passes between the light source and sensor, the light level reaching decreases in an irregular manner. Similar to the bouncing switch, the irregular decrease of light level can be detected as two or more events rather than a single event. Most physical objects blocking a light source will produce a signal which increases or decreases in a nonmonotonic manner.

The problem of an irregularly decreasing and increasing light signal can be dealt with by using a Schmitt trigger. The Schmitt trigger reduces the risk of sudden reversal of the light level state by changing the threshold light-level signal. For example the sensor may be adjusted to produce 3 V when the light is not blocked and 1 V when blocked. The threshold between blocked and unblocked could then be defined as 2 volts for example. Once the light level goes below 2 V, one doesn't want a transient fluctuation to 1.99 V to be interpreted as unblocking. Therefore when the light signal goes below 2 V, the threshold for unblocking to be detected could be set to 2.5 V. so that it it unlikely that a spurious fluctuation in a decreasing light level could be interpreted as two people passing through the door. Another way of stating this is to say the the unblocked-blocked transition occurs at 2 V whereas the blocked-unblocked transition occurs at 2.5 V.

The Schmitt trigger is usually designed using a comparator or op-amp. The following circuit shows one possible way to use the LM311 comparator to create a Schmitt trigger for use with the CdS photo resistor. The theoretical threshold levels of this circuit can be calculated from formulas explain by Malvinos's *Electronic Principles* or other electronics textbook. Remember that the output of is open collector. The actual threshold levels are usually slightly different from the theoretical ones and can be easily measured. The 5 k potentiometer allows you to adjust the voltage level received by the inverting input of the op-amp and usually needs to be adjusted for different lighting situations.

Determine the actual threshold levels of your Schmitt trigger and adjust so that you can detect shadows passing over the CdS cell. Then write a program to count the number of times a shadow passes when the output of the LM311 is connected to one of the Stamp's input pins. [In a room lit diffusely with fluorescent lighting, you may find it more effective to shade the photocell by cupping your hand over it.]

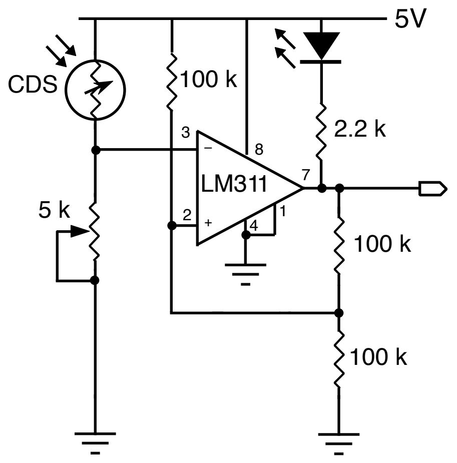

Fig 3: The Schmitt trigger enables the passing shadows to be reliably detected without false reversals. The CdS photo-resistor can be replaced with a phototransistor or a photodiode, with appropriate adjustments in the other components.

4. Counting Events

The PBasic command COUNT allows you to count the number of pulses within a given time interval. The syntax of the count command is

```
COUNT Pin, Period, Variable
```

where Pin is the pin number on which pulses are being counted, Period is the time period during which the pin is monitored in ms from 1 to 65545, and Variable is a 16-bit variable to receive the result.

Connect the output of the Schmitt trigger to an unused pin of the Stamp. Write a program loop which counts for 1 second intervals and continually displays a cumulative total to the screen which is updated every second. See the example program for the COUNT command in the *Basic Stamp Manual* (basicstampman.pdf). The suggestion of using a logic chip with a Schmitt trigger input (74HCT14) is valid; however, building your own Schmitt trigger from a comparator has the advantage of allowing you to vary the trigger levels to your own specifications.

5. (optional)The 74LS161 counter/timer

Counting events with the micro-controller involves continually looping and monitoring the input pin. This keeps the micro-controller occupied with a very simple task that could be with a much simpler device. One could, for example, connect the 74LS161 counter you used in an previous lab. The output of the Schmitt trigger should be connected to the clock input of the 74LS161 and the other input and output pins can be connected to the Stamp I/O pins.

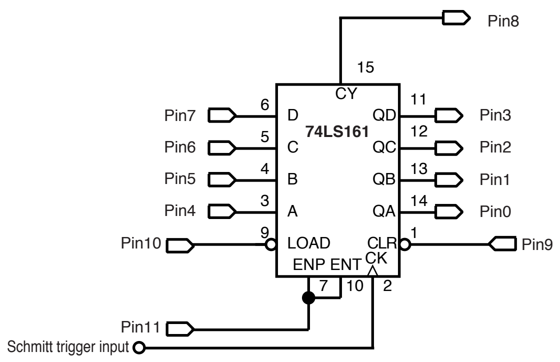

Fig 4: An external counter chip such as the 74LS161 is useful to keep track of events while the micro-controller is occupied by other tasks.

Write a program to set up the counter to signal through the carry bit when 10 events have occurred. It would be slightly better if the carry output set a flip-flop which could be cleared when detected by the program as in the “handshaking” exercise in lab 3. Connect such a flip-flop and rewrite the program accordingly. With this arrangement the micro-controller needs to periodically check the carry status and can be occupied with other tasks as well while events are being counted.

6. (optional) The Programmable counter/timer

The programmable 82C54 (or nearly equivalent 8253) counter/timer chip provides three 16-bit counters which can be programmed to operate in various modes. The three counters can be cascaded so in principle a 48-bit counter can be configured. The counters can be programmed to operate in BCD mode as well as binary. BCD mode may be useful if decimal output is needed instead of binary (with the limitation of a maximum counter of 9999 instead of 65535 for each 16-bit counter.)

The 82C54 interface to the micro-controller has a parallel 8-bit data port which must be read twice to read the value of all 16 counter bits. There are 5 other interface signals which conform to the data/address interfacing scheme of Intel microprocessors. It is possible to use this chip with the Stamp or other PIC micro-controllers; however, the internal counter/timers of the micro-controller and the large number of pins needed to interface the chip make it somewhat inconvenient to use for most applications. The figure below shows one scheme of interfacing the the chip to the Basic Stamp’s I/O pins. The protocol for using the chip can be found on the data sheet.

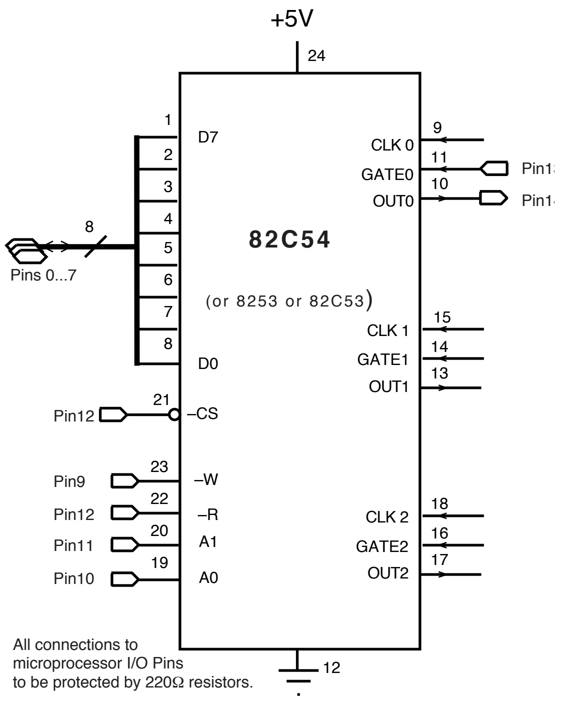

Fig 5: The 82C54 counter/timer chip has an 8-bit paralle port. This involves using many of the Basic Stamps I/O Pins but provides a very versatile device for counting or timing.

Your own ideas:

Your own ideas: