NAME (PRINT) | STUDENT ID | LAB DATE | GRADE

Experiment 9

PICLab data acquisition project

This PIC project is meant as a chance for you to put all your skills in digital/analog electronics, PIC programming/interfacing and practical problem solving to use and to build a useful device of your own. Your project will consist of the following steps:

1. research:
   the research cycle involves a preliminary exploration of possible project ideas where the hardware and software implementation options and perceived design difficulties are considered. Analyse the evolutionary steps involved in the design process that need to be successively fulfilled to make your project work; the choice of input sensors, the interpretation and processing of the input data, and the output of results or control signals that are required to give your design its planned functionality. Consider also the enhancements that, time permitting, could be made to your design to increase the usefulness or capabilities of your project;

2. proposal:
   the proposal consists of a written and oral presentation of the tentative PIC design project that you have selected. The proposal outlines what you are planning on doing, how you are planning to do it, and the difficulties that you expect to overcome. It should be the end result of some significant amount of time spent thinking about the practical issues involved in the implementation of the project, the analysis of several possible alternative hardware/software solutions to the problem, specifically in terms of the merits of the different approaches;

3. development:
   this involves the experimental process of attempting and achieving the milestones set out in the design proposal, and will likely involve a mixture of the assembly of mechanical and electronic components as well as software programming. A part of this process involves the resolution of problems, expected and unexpected, that arise during the development of any prototype system. A detailed, dated documentation of the development process must be preserved. This is a record of the steps undertaken to accomplish the end result, as well as the design choices that were attempted but proved to be unsuccessful;

4. presentation:
   as a final step, you will make an oral presentation consisting of a demonstration of the working project and a description of its operation. The presentation concludes with an assessment of your final result in terms of the goals set out in the project proposal. You should have available for submission a printed copy of your design records as well as a user manual that describes the device opeartion and includes hardware schematics and software code.

Below are several simple ideas, you are welcome to suggest a similar project of your own for approval by the instructor.

9.1 A PID temperature controller

One of the simple ways of controlling semi-continuously the state of a system is through the adjustment of the ratio of on/off times. For example, if the temperature of a reservoir is below the target temperature, we turn the heater on, and if it exceeds the target temperature, we turn the heater off. A temperature sensor and a TTL relay acting as the power switch for the heater under the control of a microprocessor is enough to implement a simple temperature controller. However, a truly intelligent controller would implement the so-called PID (Proportional-Integral-Derivative) control algorithm, whereby the temperature difference with the target temperature (P), the time integral of temperature (I) and the time rate of change of temperature (D) are all monitored and the output is controlled by all three, with varying-weight contributions. The reasoning is simple: a purely Proportional control always overshoots the target and, depending on the thermal inertia of the heater, may end up always oscillating about the target temperature. Including the other contributions (I,D) can produce an approach to the target temperature that is free of overshooting, and maintain a stable target temperature. A further improvement is to automatically adjust the relative weights of the P,I, and D contributions, to maintain this stability even if the thermal inertial properties are changing, for example as the level of the fluid in the reservoir is changing.

9.2 An ultrasonic pinger

First developed for distance measurement in autofocus cameras, ultrasonic distance measurement is based on a simple time-of-flight measurement for an ultrasonic sound pulse. It can be used to measure distances with high precision ($\pm$2mm is typical), using a standard ultrasonic transceiver (both a transmitter of the pulse, and a receiver of the echo signal from a distant object) and a timer-counter that measures the time between the sending of the pulse and the receiving of the echo. With proper calibration (the speed of sound in the air varies with temperature and humidity), a transceiver interfaced to a microcontroller is an excellent distance measuring device. It can be used in a Mechanics lab, or for non-contact level measurements in a well, or as a parking guide device in tight spaces.

9.3 A chaotic dripping tap

The incessant drip-dripping sounds of a leaky tap drive us insane late at night, but a precise measurement reveals that the time interval between drips is almost never perfectly periodic. In fact, drop formation is influenced by a number of factors, including the size of the previous drop. As a result, the dripping tap is an example of a chaotic system, and a variety of dynamic measurements can be performed, demonstrating period-doubling, attractors, and other features typical of the dynamics of chaos. An infrared LED/detector pair interfaced to a microcontroller is sufficient to perform time-between-drips measurements that can be logged and analyzed.

9.4 A universal digital knob

There is a distinct advantage to being able to "turn a knob" to control something; there is a tactile feedback that tends to feel natural to the operator, and reduces the error rate. However, analog knobs, based on variable resistors or capacitors have limited range, and are mechanically unreliable; they are also quite expensive. On the other hand, an electronically-controlled counter that uses up/down buttons is awkward to manipulate and provides no tactile feedback. One interesting compromise can be realized if a digital rotary encoder can be used to control a digital register value. In general, these devices generate a series of pulse that can be used to increase/decrease a counter value; in the best designs, some accounting of the speed of the turning is made, so that slowly turning the shaft produces individual pulses, while turning it more rapidly also increases the "step size", generating more pulses per unit rotation, a form of adaptive "ballistic" control. One of the most inexpensive ways to implement such a universal digital knob is by interfacing a stepping motor from an old floppy drive to a microcontroller.

9.5 An LCD bar-graph

A companion project to the previous one is a way to display the value of a register as a simulated bargraph, using the special block graphics characters available on an LCD display interfaced to a microcontroller. Some button action (up/down scrolling through a list of registers, or simply one-of-five single-button selection of registers) selects a particular variable/register; the "digital knob" changes its value, and the LCD displays the name/address of the variable and its value, with a simulated LCD bargraph providing a visual representation of the value. As a stand-alone project, an LCD bargraph could be used as an add-on feature to a straightforward digital voltmeter.

9.6 A joystick-controlled servomotor

Fly-by-wire controls physically separate the controller (a joystick) and the devices that are performing the action (servomotors). Instead of mechanical or hydraulic linkages, an electronic connection is made. In an analog joystick, the position controls two variable resistors (x– and y–directions) and in a digital one, it effectively presses one of four switches, one for each of x, y, $-x$, and $-y$ (or eight, if each direction has two positions, indicating half and full strength). This information is processed by a microcontroller and converted into an electronic signal that is then sent to a remote microcontroller that in turn uses it to control the state of an action device. Along the way, the command signal may be manipulated in some way, for example to ensure that an unstable or a dangerous configuration is not accidentally requested by a user, or that all transitions from one state to another are optimally smooth and do not exceed a certain rate of change (in an aircraft this might result in a dangerous-to-pilot g-force). A single microcontroller with a joystick at its input and a servomotor at its output is enough to create a simple fly-by-wire controller that demonstrates most of the essential features of such a system.

9.7 Decoding an infrared remote control

Infrared pulses can be used to carry signals wirelessly across short distances in line-of-sight configurations. These are used in a variety of devices, from TV remote controls, to shared office printers, to toys such as the Furby, capable of some form of communication with its own kind. To build or debug an IR-enabled communications device it is useful to have a decoder of what these devices are sending through the IR, into a sequence of readable ASCII codes. A more ambitious project would involve two-directional communications, both reading what the other IR transmitter is sending, and sending one's own signals out. This is easily implemented using a microcontroller; this is what is at the heart of all "universal programmable" remote controls that are widely available. With a microcontroller-based decoder/encoder attached to a serial port, you can make your computer communicate with all such devices, and to replace all of your remote controls with a software one.

9.8 A PIC-based mouse controller

A typical electromechanical pointing device (a mouse) has a large rubberized ball rolling on the work surface and in turn rotating two orthogonal slotted wheels, one for the x- and one for the y-motion. Each slotted wheel interrupts two infrared beams, producing a pair of logical pulses, 90° out-of-phase. Each pair of pulses corresponds to a step of displacement, and their relative phase determines the $\pm$ direction of motion. Typically, these TTL signals are converted by a dedicated mouse controller IC into a stream of codes sent through a serial connection to the computer, reporting the displacement of the mouse. The same task could be performed in software running on a microcontroller. In this way, non-standard display modes (for example, direction and velocity display) could be programmed. If a sufficiently fine resolution can be obtained, the two-wheel interface could be adapted to display in real time the action of a chaotic two-pendulum magnetic toy, or to the monitoring/control of a model Foucault pendulum. For the latter, useful pointers are:

- Richard Crane's article in the References section on the PHYS 2P32 website
- `http://en.wikipedia.org/wiki/Foucault_pendulum`
- `http://www.iop.org/EJ/article/0031-9120/19/6/412/pev19i6p294.pdf`
- `http://www.sas.org/E-Bulletin/2002-04-26/handsOnPhys/body.html`