

CS402 Project Phase I

Group 10: Michael Nolan

1 Arduino Capabilities

We are using an Arduino Uno SMD-Ver development board for prototyping our embedded system project, which will be discussed later. Our version of the board is using an Atmel's ATmega328 micro controller. Features of this micro can be located at [1] and also in [2] whereas a more comprehensive explanation for the terminology and functions offered can be seen at [3]. A few features offered by the chip are highlighted below:

1. 23 general purpose input/output pins are provided (only 20 are accessible on the development board however),
2. operates between 1.8-5.5v (while plugged into the USB on the development board will use 5v),
3. from the 23 GPIO pins, as many as 6 can be used as pulse-width modulated outputs for generating analogue output,
4. also from the 23 GPIO pins, 6 of these pins can be configured to be analogue voltage reading pins (not the same 6 that are used as PWMs),
5. the chip provides hardware support for universal serial asynchronous receiver transmitter (the development board uses these pins to communicate with

FTDI's chip for 'USART to USB' interfacing and hence talking to the host computer if one is connected),

6. there is hardware support for as many as 3 timers as well as SPI and I²C hardware-based interfaces,
7. there are also a small number of power-saving modes that the ATmega328 can be put into when the program is idle.

Also worth noting is that the ATmega328 has 32Kbytes of program memory, 1Kbyte of EEPROM and 2Kbytes of RAM. Since the development board has already utilized some of this for the Arduino boot loader (used to upload programs from host computers) then we can't expect to get exactly this capacity when we come to implement our project. The ATmega328 operates at a clock speed of 16MHz. Based on [4], most primitive ATmega328 instructions (ADD, SUB, JMP, CALL, etc) will take between 1 and 4 clock cycles, meaning roughly 4 million instruction per second, or 4 instructions per microsecond.

2 Idea

The idea we choose is quite simple: we will replicate a classic arcade game using an Arduino board and an lcd screen.

3 IDE

We will make use of the freely obtainable Eclipse IDE and win-avr plug in for Eclipse which will allow others to recompile the source code with little or no changes needed. This project is small enough that it shouldn't merit too much debugging if any, as we will produce a working example on the computer before porting it over to the board the only problems that should occur is in working with the LCD and ensuring that the buttons correspond to the keyboard presses.

4 Required resources

Before we go on to explain the project, below is a selection of components that will be useful in constructing it.

1. Arduino board x 1,
2. Push buttons x 6,
3. Pull-up resistors x 6 (value 10k ohm preferred),
4. Display 128x64 x 1.

5 Time plans

The list below shows the plan for the construction of the project, it is estimated that each of these points should take no more than a few hours, if even.

1. Confirm working-ness of display and can be properly used (i.e can we write a pixel to the right place on the screen),
2. Confirm we can check for input via the push buttons,

3. Research 'retro' games to find a suitable example,
4. Construct (and hence simulate) this example on the computer,
5. Port example onto arduino board.

6 Design

We feel that this will be better reflected in the PhaseII documentation, at which time the decision on which retro game to be used will have been picked. The general outlook and feel of the project is that it should be a simple and easy to understand as possible, as such we have already considered restricting the number of buttons to as few as possible - we ask for 6, but we expect that the game we choose will not require all of those.

Since the code will probably be read by others, we will try our best to ensure that it is both readable and very simple although we note that since we are designing for a MCU as opposed to a GPCPU then we will likely do small optimisations which may confuse others, we will do our best to note these where appropriate.

References

- [1] http://www.atmel.com/dyn/products/product_card.asp?part_id=4720.
- [2] http://www.atmel.com/dyn/resources/prod_documents/8271S.pdf.
- [3] http://www.atmel.com/dyn/resources/prod_documents/doc8271.pdf.

[4] http://www.atmel.com/dyn/resources/prod_documents/doc0856.pdf.