Embedded Software

CS 145/145L

Caio Batista de Melo
Recap

Project 1

- built power source;
- powered microcontroller;
- connected programmer and microcontroller;
- controlled external device (LED) from code;
- read input from external device (button) in code;
- added external crystal for stable/precise timing.
Design an embedded computer centered around the ATMega32 microcontroller. For input: use a keypad; For output: use an LCD;

Write a C program that implements a digital clock, showing date (MM/DD/YYYY or YYYY-MM-DD) on the top row and HH:MM:SS (12h or 24h is fine) on bottom row. Provide all the UI needed to set the date and time on your digital clock.

https://canvas.eee.uci.edu/courses/45047/assignments/929270
Project 2 Roadmap

Keypad → LCD → Application Logic
Keypad Rough Layout
Why use a keypad?

- Schematic of one push button
- It will take a lot of push buttons to design a keypad
- Since there are 16 keys, 16 GPIOs will be used logically which is a lot of resources (pins)
Keypad Internals
PORT B is needed for ISP and PORT A is needed for future projects. So we need to choose from PORTs C and D; let’s save PORT D for LCD.
DDR and PORT Logic (Quad-State GPIO)

Read is done through the PIN Register.
Test Your Understanding

DO **NOT** DO THIS!
Test Your Understanding
Test Your Understanding

PB0
“W1”
AVR
PIN
Read ?

SW1
What is the state if we read this pin?

What is the state now?

What is the state if we read this pin?
Checking for Key Press

```c
int get_key() {
    int i, j;
    for (i=0; i < 4; i++) {
        for (j=0; j < 4; j++) {
            if (is_pressed(i, j)) {
                return 4 * i + j + 1;
            }
        }
    }
    return 0;
}
```

```c
int is_pressed(int r, int c) {
    // Set all 8 GPIOs to N/C
    DDRC = 0;
    PORTC = 0;
    // Set r to "0"
    // Set c to "w1"
    if (/* value of c == 0 */) {
        return 1;
    }
    return 0;
}
```

What if multiple keys are pressed?
Should have a method that gives 16 different combinations for different keys that were pressed!

Ideas?
- One LED for each key;
- Turn LED on for $key$ seconds;
- Blink LED $key$ times;
- …

```c
int main() {
    while (1) {
        avr_wait(1000);
        int i, k;
        k = get_key();
        for (i=0; i < k; i++) {
            // Turn an LED on.
            avr_wait(500);
            // Turn the LED off.
            avr_wait(500);
        }
    }
    return 0;
}
```
JTAG (Joint Test Action Group)

Disable JTAG Programmer on fuse settings.
Learn to solder :) 
- https://www.youtube.com/watch?v=QKbJxytERvg
- https://www.youtube.com/watch?v=oRt_jOJ8IRU
- https://www.youtube.com/watch?v=k4lDMfMIoIu

Feel free to use the soldering iron in the lab; You can ask your TAs for help!
LCD Pinout Diagram
See you next time :)