CSC270 Spring 2016

Circuits and Systems
Lecture Notes, Week 12

Dominique Thiébaut
dthiebaut@smith.edu
C Continued

The Arduino
Exercise

Write a C program that declares a string variable containing "Smith College" and replaces all the characters by the first letters of the alphabet. Your program must use a loop!
#include <stdio.h>
#include <string.h>

//-- new type --
struct Pair {
    int first;
    int second;
};

void printPair( struct Pair p ) {
    printf( "(%d, %d)" , p.first, p.second );
}

int main() {
    struct Pair p1, p2;

    p1.first = 123;
    p1.second = 45;

    p2 = p1;
    printPair( p1 );
    printPair( p2 );

    return 0;
}
#include <stdio.h>
#include <string.h>

//--- new type ---
struct College {
    char* name;
    char rating;
    int pop;
    float endow;
};

typedef struct College CollegeStruct;

void printCollege( CollegeStruct cs ) {
    printf( "(%s, %c, %d, %1.2f)\n", 
        cs.name, cs.rating, cs.pop, cs.endow );
}
```c
int main() {
    CollegeStruct col1;
    col1.name = "Smith College";
    col1.rating = 'A';
    col1.pop = 2400;
    col1.endow = 1800; // millions
    printCollege( col1 );
    return 0;
}
```
Outline

The Arduino

Introduction

The ATMega168 MCU

Arduino Schematics

Power

Programming/IDE

Digital I/O Pins (Output, PWM, Input)

Analog I/O Pins
The Arduino
Why?

- **Natural progression:**
  - transistor
  - → gate
  - → comb. & seq. circuits
  - → microprocessor
  - → I/O ports
  - → *microcontroller* + C
References

  - Good for installation of software on Windows and Mac

- **Arduino's Site**: [https://www.arduino.cc/](https://www.arduino.cc/)

- **LadyAda**: [http://www.ladyada.net/](http://www.ladyada.net/)
C Continued

The Arduino

Introduction

The ATMega168 MCU

Arduino Schematics

Power

Programming/IDE

Digital I/O Pins (Output, PWM, Input)

Analog I/O Pins
The Arduino

https://www.arduino.cc/en/Main/arduinoBoardDiecimila
The Arduino

ATmega168

https://www.arduino.cc/en/Main/arduinoBoardDiecimila
We stopped here last time...
ATMega168

- Made by **Atmel**, microchip manufacturer (automotive, spacecraft, sensors, MCUs)
- **8-bit** processor
- **16 KBytes** of Flash Memory (**Arduino uses 2 KB for boot-loader**)
- **512 Bytes** EPROM
- **1KBytes** RAM
- **Write/erase cyles**: 10,000 flash/100,000 EEPROM
- **Data retention**: 20 years
- **32 registers/131 instructions**
- **20 MIPS @ 20 MHz** (**Arduino works at 16 MHz**)

ATMega168

• Operating voltage: 1.8-5.5V (Arduino uses 5V)

• Input voltage: 7-12V

• 6 Analog input pins (10 bits)

• 14 Programmable digital I/O pins

### N-Queens

Execution times for N-Queens program (C or Java) in milliseconds

![Table of execution times for N-Queens program](image)

<table>
<thead>
<tr>
<th>NxN</th>
<th>Macbook Pro 2009</th>
<th>Macbook Pro 2014</th>
<th>MacPro 2009</th>
<th>MacPro 2014</th>
<th>Linux Mint Beowulf2</th>
<th>Laptop 1</th>
<th>Linux 8</th>
<th>Arduino</th>
</tr>
</thead>
<tbody>
<tr>
<td>8x8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>9x9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>10x10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>11x11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>12x12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>16x16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>17x17</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>29x29</td>
<td>1518</td>
<td>33667</td>
<td>12586</td>
<td>23558</td>
<td>4069</td>
<td>244</td>
<td>9041</td>
<td>17252</td>
</tr>
</tbody>
</table>

ATMega168 Architecture

executed in every clock cycle. The program memory is In-System Reprogrammable Flash memory.

The fast-access Register File contains $32 \times 8$-bit general purpose working registers with a single clock cycle access time. This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typical ALU operation, two operands are output from the Register File, the operation is executed, and the result is stored back in the Register File – in one clock cycle.

Figure 8-4. On-chip data SRAM access cycles.
• Bit 0 – IVCE: Interrupt vector change enable

The IVCE bit must be written to logic one to enable hardware four cycles after it is written or when IVS interrupts, as explained in the IVSEL description above. See code example below.

Assembly code example

```
Move_interruption:
    ; Get MCUCR
    in r16, MCUCR
    mov r17, r16
    ; Enable change of Interrupt Vectors
    ori r16, (1<<IVCE)
    out MCUCR, r16
    ; Move interrupts to Boot Flash section
    ldi r17, (1<<IVSEL)
    out MCUCR, r17
    ret
```

C code example

```
void Move_interruption(void)
{
    uchar temp;
    /* Get MCUCR*/
    temp = MCUCR
    /* Enable change of Interrupt Vectors */
    MCUCR = temp|(1<<IVCE);
    /* Move interrupts to Boot Flash section */
    MCUCR = temp|(1<<IVSEL);
}
```

This bit is not available in Atmel ATmega48.
The Arduino

Introduction

The ATMega168 MCU

Arduino Schematics

Power

Programming/IDE

Digital I/O Pins (Output, PWM, Input)

Analog I/O Pins
The Arduino


Photograph by SparkFun Electronics. Used under the Creative Commons Attribution Share-Alike 3.0 license.

https://www.arduino.cc/en/Main/arduinoBoardDiecimila
The Arduino

Introduction

The ATMega168 MCU

Arduino Schematics

Power

Programming/IDE

Digital I/O Pins (Output, PWM, Input)

Analog I/O Pins
Power

- +5V
- +7÷12V
- 3.3V 50mA
- 5V
- 6-12V
How **Long** can a **9V** Battery Power the **Arduino**?

- Hard to tell: depends on what's connected

- **Solution**: make Arduino monitor itself (with analog input)!

C Continued

The Arduino

Introduction

The ATMega168 MCU

Arduino Schematics

Power

Programming/IDE

Digital I/O Pins (Output, PWM, Input)

Analog I/O Pins
D. Thiebaut, Computer Science, Smith College
```c
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
*/

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);    // set the LED on
  delay(1000);               // wait for a second
  digitalWrite(13, LOW);     // set the LED off
  delay(1000);               // wait for a second
}
```
The general idea

```c
void setup() {
    // initialize the digital pin as an output.
    // Pin 13 has an LED connected on most Arduino boards:
    pinMode(13, OUTPUT);
}

void loop() {
    digitalWrite(13, HIGH); // set the LED on
    delay(3000); // wait for a second
    digitalWrite(13, LOW); // set the LED off
    delay(3000); // wait for a second
}

int main() {
    setup();
    while (1) {
        loop();
    }
}
```
Getting Started with Arduino

Learn Arduino

Read an introduction on what is Arduino and why you'd want to use it.

Exercise

Follow the tutorial and make the on-board LED blink

Getting Started with Arduino

Learn Arduino

- Read an introduction on what is Arduino and why you'd want to use it.
Serial Output

1. set the baud rate (9600 baud) in IDE
2. set the baud rate in `setup()` function
3. print away!
void setup() {
    Serial.begin(9600);
    Serial.println("Sketch started!");
}

void loop() {
    ...
    Serial.print("a = ");
    Serial.println(a);
    ...
}
C Continued

The Arduino

Introduction

The ATMega168 MCU

Arduino Schematics

Power

Programming/IDE

Digital I/O Pins (Output, PWM, Input)

Analog I/O Pins
Digital I/O Pins

https://s-media-cache-ak0.pinimg.com/736x/f9/cf/ec/f9cfec14e2ffa4820ec8ec7c5044ee73.jpg
Digital I/O Pins

• **Digital** != Analog
Digital I/O Pins

• **Digital** != Analog

• **14** Pins, programmable as **Output** or **Input**
  (Note: Pin 13 already connected to on-board LED)
Digital I/O Pins

• **Digital** != Analog

• **14** Pins, programmable as **Output** or **Input**
  (Note: Pin 13 already connected to on-board LED)

• Internal **pull-up** 20KOhm resistor for input mode. Can be programmed to be ON or OFF.
Digital I/O Pins

- **Digital** != Analog

- **14** Pins, programmable as **Output** or **Input**
  (Note: Pin 13 already connected to on-board LED)

- Internal **pull-up** 20KOhm resistor for input mode. Can be programmed to be ON or OFF.
Simple **Steps** To Follow

1. **Pick a Digital Pin** that is unused
2. **Connect** Hardware to it
3. **Program**:
   1. *Setup()*: Set direction of pin
   2. *Loop()*: output to, or input from pin
Output Pin: Example

- **pinMode()**
- **digitalWrite()**

```cpp
int ledPin = 13; // LED connected to digital pin 13

void setup() {
    pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop() {
    digitalWrite(ledPin, HIGH); // sets the LED on
    delay(1000); // waits for a second
    digitalWrite(ledPin, LOW); // sets the LED off
    delay(1000); // waits for a second
}
```

(Taken from the "Blink" example in Arduino sketches)
Arduino-Related C Constants

- **INPUT**
- **INPUT_PULLUP**
- **OUTPUT**
- **HIGH**
- **LOW**

*To setup a pin as input or output*

*To set output pin, or to test input pin*
Important Rule

Once the behavior of a pin is selected (in, out, pwm...), DO NOT change it during the life of the project!
C Continued

The Arduino

Introduction

The ATMega168 MCU

Arduino Schematics

Power

Programming/IDE

Digital I/O Pins (Output, PWM, Input)

Analog I/O Pins
Additional I/O Functionality: Pulse-Width Modulation (PWM)
Additional I/O Functionality: PWM
Additional I/O Functionality: PWM
Additional I/O Functionality: PWM
Additional I/O Functionality: PWM

Fan
ON
Additional I/O Functionality: PWM
Additional I/O Functionality: PWM

Wanted
Additional I/O Functionality: PWM
PWM pins = 3, 5, 6, 9, 10, and 11

```c
int pinPWM = 3;

analogWrite(pinPWM, 0);

analogWrite(pinPWM, 64);

analogWrite(pinPWM, 127);

analogWrite(pinPWM, 191);

analogWrite(pinPWM, 255);
```

from https://www.arduino.cc/en/Tutorial/PWM

490Hz

(no need to set pin in output mode)
The Arduino

Introduction

The ATMega168 MCU

Arduino Schematics

Power

Programming/IDE

Digital I/O Pins (Output, PWM, Input)

Analog I/O Pins
**Input**: Pin 2

**Output**: Pin 13 (LED)

```cpp
int inPin = 2;
int outPin = 13;

void setup() {
    pinMode(inPin, INPUT_PULLUP);
    pinMode(outPin, OUTPUT);
}

void draw() {
    int switch;

    switch = digitalRead(inPin);
    digitalWrite(outPin, switch);
}
```

**Version 1: Internal pull-up**
Input: Pin 2
Output: Pin 13 (LED)

```c
int inPin = 2;
int outPin = 13;

void setup() {
    pinMode( inPin, INPUT );
    pinMode( outPin, OUTPUT );
}

void draw() {
    int switch;
    switch = digitalRead( inPin );
    digitalWrite( outPin, switch );
}

Version 2: External pull-up
```
What value of $R$?
Properties of Pins Configured as INPUT_PULLUP

There are 20K pullup resistors built into the Atmega chip that can be accessed from software. These built-in pullup resistors are accessed by setting the pinMode() as INPUT_PULLUP. This effectively inverts the behavior of the INPUT mode, where HIGH means the sensor is off, and LOW means the sensor is on.

Read the Documentation!
The Arduino

Introduction

The ATMega168 MCU

Arduino Schematics

Power

Programming/IDE

Digital I/O Pins (Output, PWM, Input)

Analog I/O Pins
Principles of Analog to Digital Conversion (ADC)
### 4-to-2 Priority Encoder

<table>
<thead>
<tr>
<th>I0</th>
<th>I1</th>
<th>I2</th>
<th>I3</th>
<th>Y1</th>
<th>Y0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>X</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
2-bit ADC
4 V → 2 bits
- 10-bit ADC
- 1024 discrete levels
- 5V max input voltage (but we can go around the limit)

\[
\frac{\text{Input Voltage}}{5 \text{ V}} = \frac{\text{ADC Output}}{1023}
\]
Example

Taken from https://www.arduino.cc/en/Tutorial/AnalogInput
Example
int sensorPin = A0;   // select the input pin for the potentiometer
int ledPin = 13;      // select the pin for the LED
int sensorValue = 0;  // variable to store the value coming from the sensor

void setup() {
    // declare the ledPin as an OUTPUT:
    pinMode(ledPin, OUTPUT);
}

void loop() {
    // read the value from the sensor:
    sensorValue = analogRead(sensorPin);
    // turn the ledPin on
    digitalWrite(ledPin, HIGH);
    // stop the program for <sensorValue> milliseconds:
    delay(sensorValue);
    // turn the ledPin off:
    digitalWrite(ledPin, LOW);
    // stop the program for for <sensorValue> milliseconds:
    delay(sensorValue);
}
Communication
Communication

Sender  Receiver
Rule #1: Ground Sender and Receiver!
Serial Communication

Sender

Receiver
Timing: Serial Communication

Start b0  b1  b2  b3  b4  b5  b6  b7  Stop

Sender Algorithm

Receiver Algorithm
Parallel Communication

Sender

Strobe →
Ack ←

D0 →
D1 →
D2 →
D3 →

Receiver

Strobe →
Ack ←

D0 →
D1 →
D2 →
D3 →
Timing: Parallel Communication

Strobe

Ack

D0-3

Time

Time

Time