LAB1: Intro to ARM assembler on R PI 3

Demo Due: **By the End of lab section on Sept 15th (50%)**

Report Due: 10:00a.m. Sept 11st (50%)

In this lab, you will learn how to programming Raspberry Pi 3 by ARM assembler and then programing GPIO pin as the output to control a LED.

Learning Reference: "ARM assembler in Raspberry Pi" Roger Ferrer Ibáñez: http://thinkingeek.com/2013/01/09/arm-assembler-raspberry-pi-chapter-1/

1. Your first program

In this section, you will follow a step by step instruction to create your very first assembly program to perform a simple arithmetic with integer variables, you do not need to demo this to your TA. If you have not installed the OS yet. Please refer to **Lab 0** document to finish all pre-installation on RPi 3.

Create a file named **helloworld.s** and edit the file with text editor or command line window. First we need to create data types for adding and the output string.

```
        .data
        .balign 4
string: .asciz "a + b = %d\n"
a:      .word   33
b:      .word   44
c:      .word   0
```

We start the program with a main function:

```
.text
.global main
.extern printf
main:
        push    {ip, lr}  @ push return address + dummy register
```

Then we get address of **a** into register **r1**, and then put the value stored in that address into **r1**. We do the same thing for **b:**

```
ldr     r1, =a
ldr     r1, [r1]
ldr     r2, =b
ldr     r2, [r2]
```

Then we add **r1** to **r2** and store the result into **r1**.

```
add     r1, r1, r2
```

Next step, we get the address of **c** into **r2**, and we copy the value in **r1** into **c**

```
ldr     r2, =c
str     r1, [r2]
```

Then we print out the result, and return the address into pc

```
ldr     r0, =string
ldr     r1, [r2]
bl      printf          @ print string
pop     {ip, pc}
```

2. Compile and Run

- Assemble the code into a object file:

```
as -o helloworld.o helloworld.s
```

- Compile the object file by a C compiler, it will be transformed into a executable:

```
gcc -o helloworld helloworld.o
```

- Run your first program!

```
/helloworld
```

3. LED Blinking

In this section you are going to write a program using ARM assembler that will blink an external LED on the breadboard. Demo to your TA in the lab.

Attach your source code attached to the end of the report with **inline comments** explain the code. You do not need to comment line by line, just present your understanding of your own work.

3.1 A C equivalent program is provided here:

```
//https://www.raspberrypi.org/forums/viewtopic.php?f=33&t=23090
#include <wiringPi.h>
#include <stdio.h>
#include <stdlib.h>

int main ( void ) {
  int pin = 7;

  if (wiringPiSetup() == -1) {
      exit (1);
  }
  pinMode(pin, OUTPUT);

  int i;
  for ( i=0; i<10; i++ ) {
    digitalWrite(pin, 1);
    delay(200);

    digitalWrite(pin, 0);
    delay(200);
  }
  return 0;
}
```

3.2 Setup PinMode in assembler:

```
init:
        ldr     r0, =pin
        ldr     r0, [r0]
        mov     r1, #OUTPUT
        bl      pinMode
```

3.3 Digital Write in assembler:

```
              .data
        pin:  .int 7
              .main
        ldr     r0, =pin
        ldr     r0, [r0]
        mov     r1, #1
        bl      digitalWrite
```

3.4 Delay(200) in assembler:

```
      .data
 delay_time: .int 200
      .main
ldr     r0, =delay_time
ldr     r0, [r0]
bl      delay
```

3.5 How to connect a LED(Pin is different, refer to 3.6):

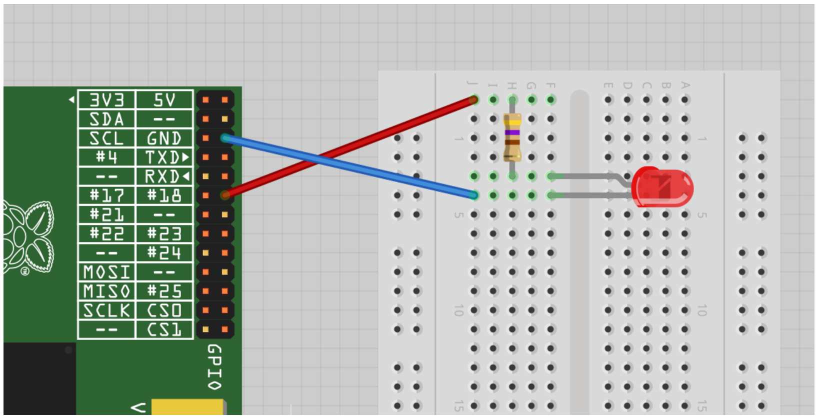

Image source:http://razzpisampler.oreilly.com/ch03.html

3.6 Raspberry Pi GPIO:

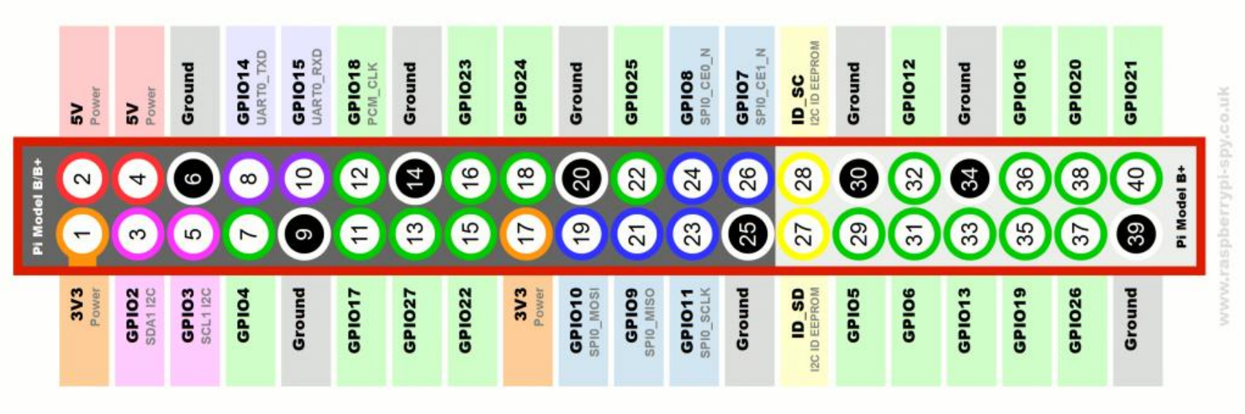

Image Source:http://www.raspberrypi-spy.co.uk/2012/06/simple-guide-to-the-rpi-gpio-header-and-pins/

4. Thinking and exploring

We‘ve prepared three question for you to think and explore about raspberry Pi. Please write down your thought to the questions in the report:

a. In this lab, we used assembler to control the output to the GPIO pin on the raspberry Pi in order to blink the LED. Do you have an alternative way to blink this external LED? If so, write down your answer.

b. What you can add/change to the code in section 3.2 – 3.3 if you want to control the LED on/off time to different numbers? (For example LED on for 3 second and off for 4 second, repeat.) Can we replace function Delay in our code?

c. Now, if we replace the external LED in this lab by a buzzer. Can we still produce music by this buzzer using the same way we do to the LED? If yes, provide your solution. If no, what needs to be change in order to produce music?