MICROPROCESSORS A (17.383)

Fall 2010

Lecture Outline

Class # 04

September 28, 2010

Dohn Bowden

Today's Lecture

- Syllabus review
- Microcontroller Hardware and/or Interface
- Programming/Software
- Lab
- Homework
- Finish Lab # 2

Course Admin

Administrative

- Admin for tonight ...
  - Syllabus Highlights
    - Exam # 1 ... next week (October 05, 2010)
    - Lab # 1 ... is due tonight
    - Lab report for Lab # 2 is due October 19, 2010
    - **No Class on Tuesday October 12, 2010**
      - **The 12th is a Monday Class Schedule**

Syllabus Review

Week	Date	Topics	Lab	Lab Report Due
1	09/07/10	Intro, Course & Lab Overview, Microcontroller Basics	1	
2	09/14/10	PIC16F684 Overview and General Input/Output	1 con't	
3	09/21/10	Switches	2	
4	09/28/10	Seven Segment LEDs	**2 con't**	1
5	10/05/10	**Examination 1**		
X	**10/12/10**	**No Class – Monday Schedule**		
6	10/19/10	Analog to Digital Conversion	3	2
7	10/26/10	Analog to Digital Conversion con't	3 con't	
8	11/02/10	LCD Interface and Assembly Language	4	
9	11/09/10	Comparators	4 con't	3
10	11/16/10	Timers and Pulse Width Modulation (PWM)	5	
11	11/23/10	Mixed C & Assembly Programming/Course Project	Project	4
12	11/30/10	**Examination 2**		
13	12/07/10	Course Project	Project	5
14	12/14/10	Final Exam/Course Project Brief and Demonstration	Demo	

Microcontroller Hardware and / or Interfaces

PIC16F684 Interfacing

- Voltage Regulator Circuit
- Seven-Segment LEDs

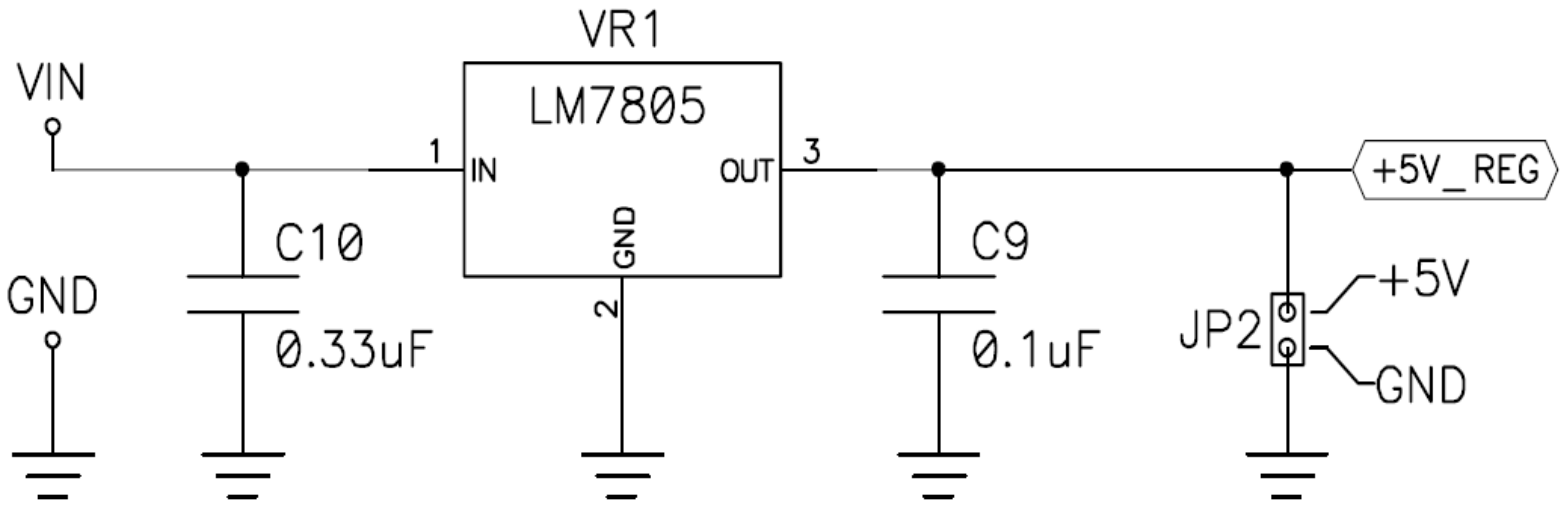
Transition From the PICkit 1 Development Board

- Thus far we have been using the PICkit board
  - All required circuitry was on the board
  - However ... the board was limited by it's design and components
- We want to be able to design our own circuits
  - Would be cost prohibited to use the PICkit as our backbone

Voltage Regulator Circuit

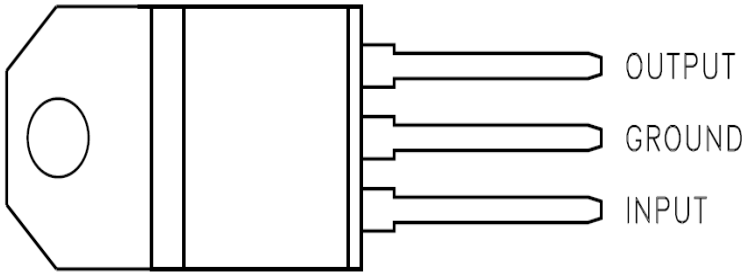
- First consideration ... power requirements
  - Need to power the PIC16F684 and its external interface components

Voltage Regulator Circuit



LM7805 Pin Connections - Top View

LM7805 Pin Connections - Top View



Breadboard

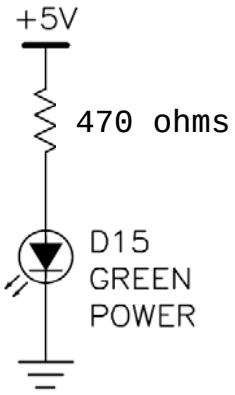
- Locate your circuit near the breadboard terminals
  - Will be using this circuit for all our projects
  - Therefore ... use minimal space on the board
- Keep the wiring as short as possible
  - Less EMI issues
- Test your circuit before you install sensitive components

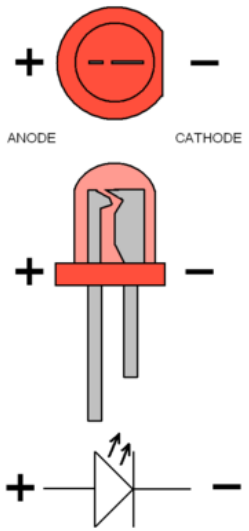
Testing Your Voltage Regulator

- Voltage checks
- Oscilloscope

Power Indicator Circuit

- PIC16F684 *MUST NOT* be removed or installed while energized
- Therefore ... add an LED for a visual indication that power is present



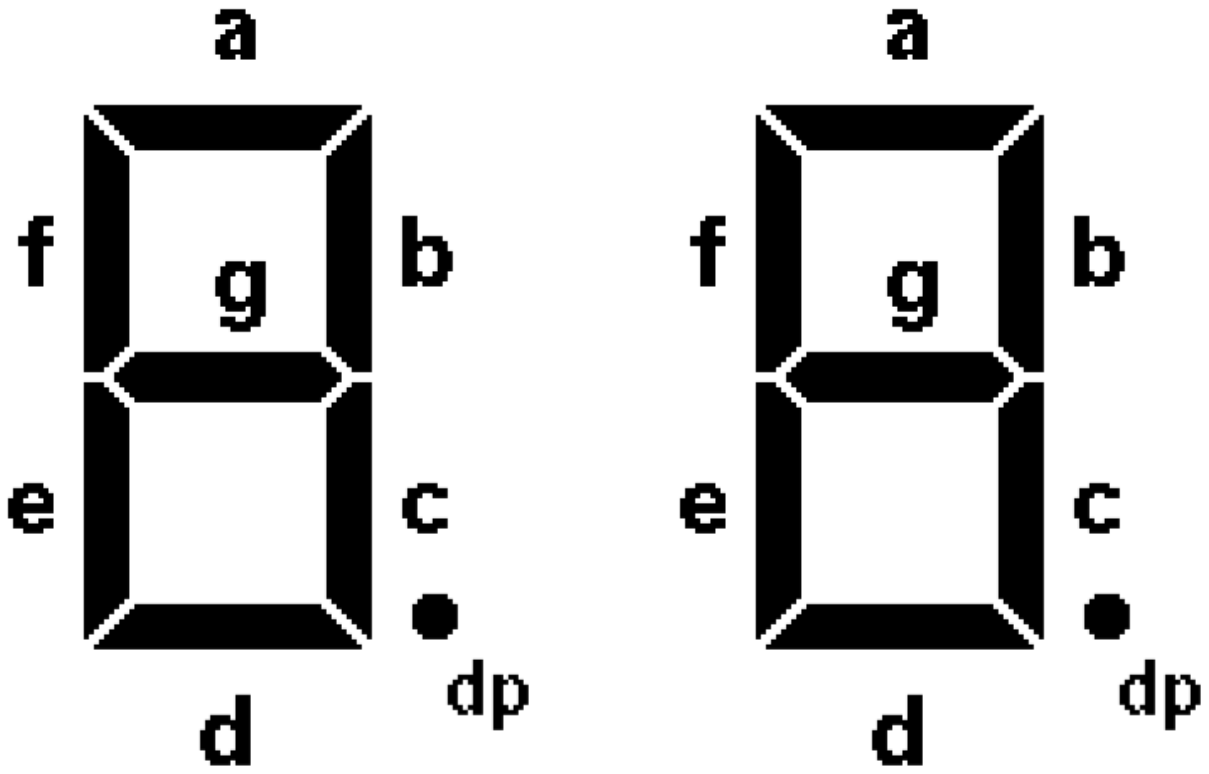


External Interface Components

- External interface components ...
  - A breadboard will allow for additional components not available on the PICkit 1
    - Additional Switches
    - Additional LEDs
    - Seven-Segment LEDs
    - LCD Displays
    - Etc.

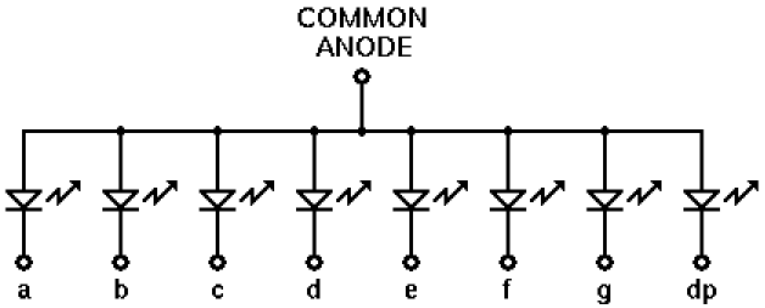
The Seven-Segment LED ...

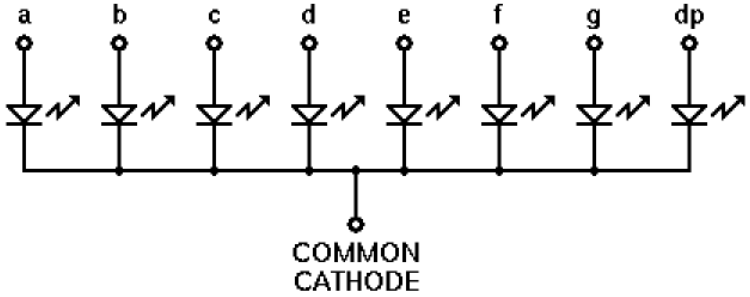
The Seven-Segment LED



The Seven-Segment LED

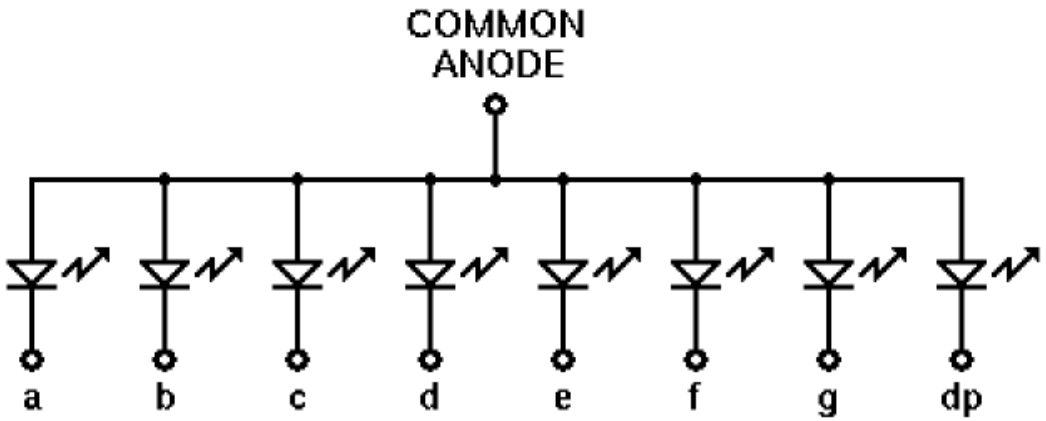
- LEDs in seven-segment display are not isolated from each other
- Either all of the cathodes, or anodes, are connected together
- The other end of each LED is individually available
- Means fewer electrical connections
  - allows us to easily enable or disable a particular digit by controlling the common lead





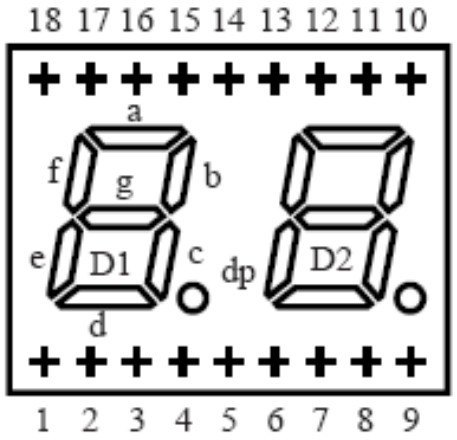
The Seven-Segment LED – Our Labs

- We will be using the *common anode* configuration



The Seven-Segment LED

Terminal Connection



Pin No.	Assignment	Assignment
1	Cathode e1	Anode e1
2	Cathode d1	Anode d1
3	Cathode c1	Anode c1
4	Cathode dp1	Anode dp1
5	Cathode e2	Anode e2
6	Cathode d2	Anode d2
7	Cathode g2	Anode g2
8	Cathode c2	Anode c2
9	Cathode dp2	Anode dp2
10	Cathode b2	Anode b2
11	Cathode a2	Anode a2
12	Cathode f2	Anode f2
13	Common Anode D2	Common Cathode D2
14	Common Anode D1	Common Cathode D1
15	Cathode b1	Anode b1
16	Cathode a1	Anode a1
17	Cathode g1	Anode g1
18	Cathode f1	Anode f1

Lighting Each Digit

- To light each digit separately, we would need 14 I/O lines on the microcontroller
- This does not include any other I/O lines needed for items such as switches
- Therefore, due to the above hardware limitations
  - We will only light one digit at a time, with a process called ***strobing***
  - ***Strobing*** 7-segment displays basically involves displaying a number on one display for a short while, and then turning off that display while you display another number on another display

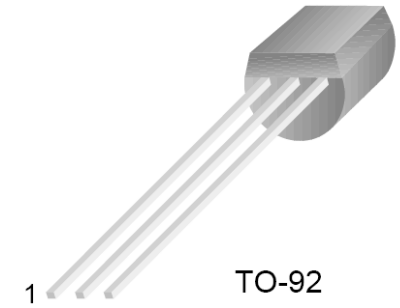
Lighting Each Digit

- The displays are in fact flashing on and off at a high rate of speed, giving the impression that they are constantly on
- To avoid flickering ...
  - Each digit must be on at least 50 times per second
- We will use an electronic switch to select the digit ...
    - A transistor

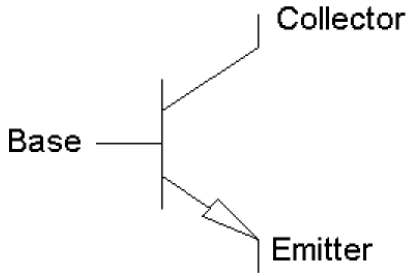
Transistors

- The easiest way to understand transistors is to think of them as switches
- You can switch a big current ...
  - between the collector and emitter
- With a much smaller current (in the base)

Transistors



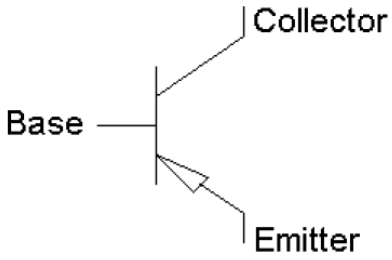
NPN transistor



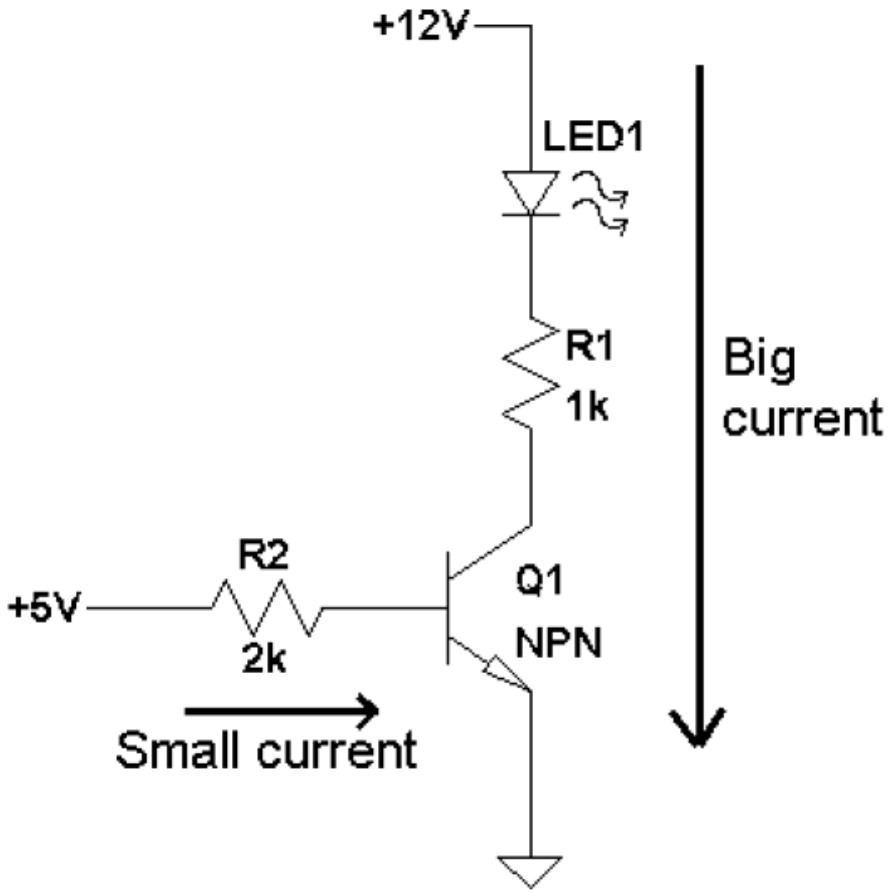
One mnemonic device for identifying the symbol for the NPN transistor is "**n**ot **p**ointing i**n**."

One mnemonic device for identifying the symbol for the PNP transistor is "**p**ointing i**n**."

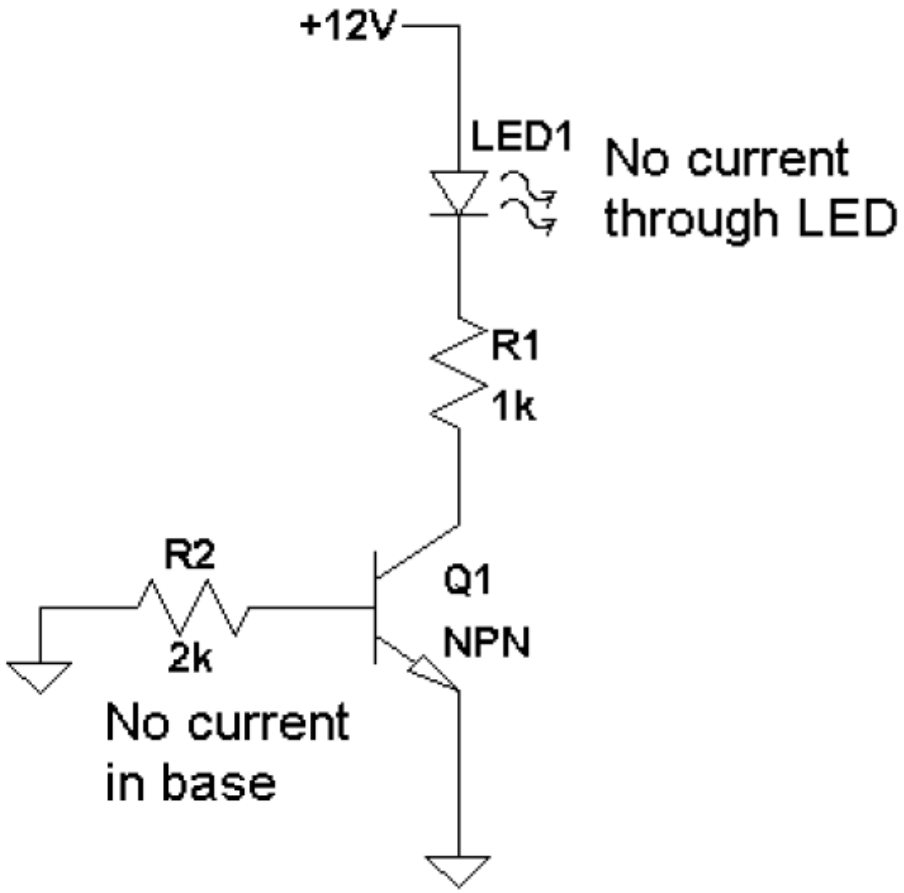
PNP transistor



NPN transistor as a switch (on)



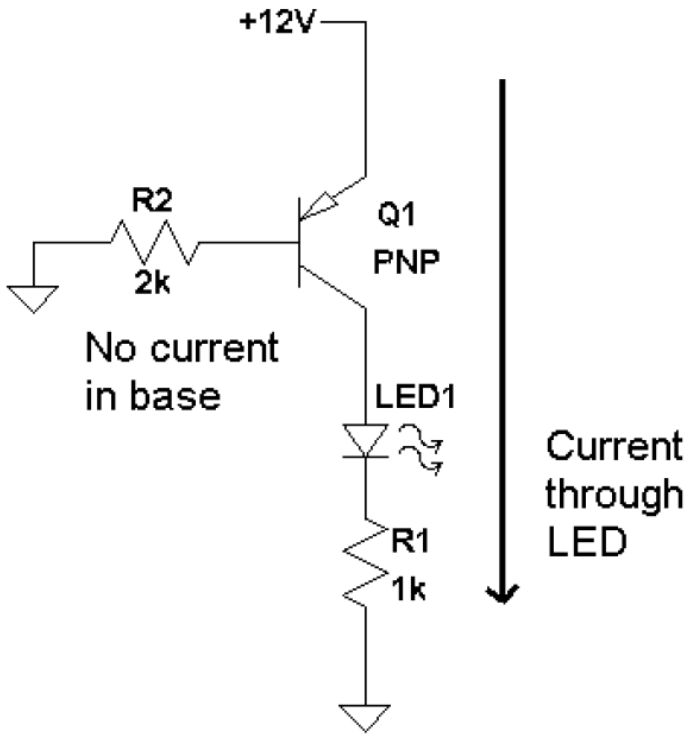
NPN transistor as a switch (off)



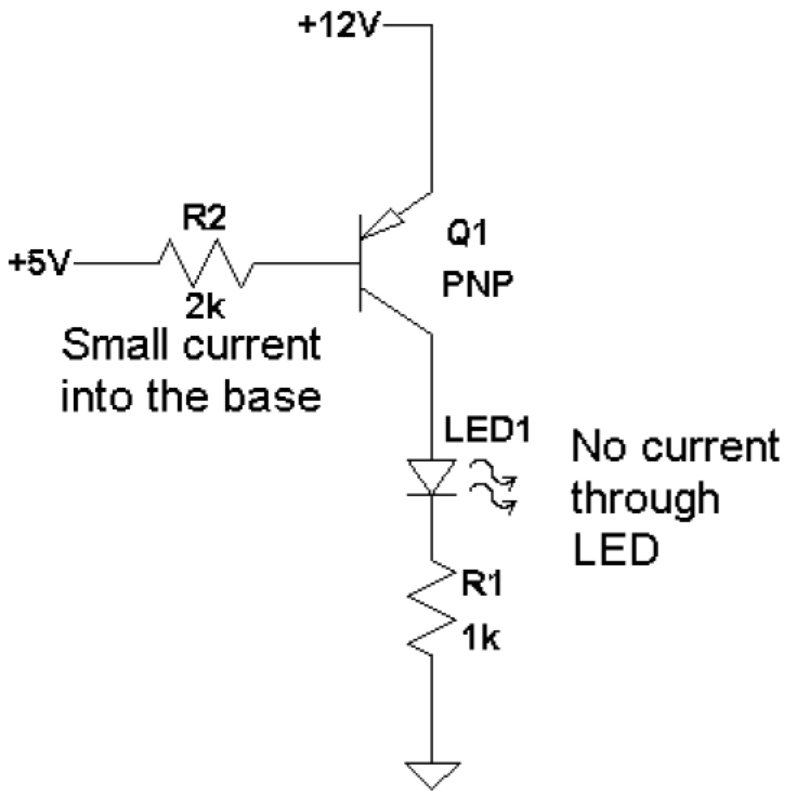
PNP Transistors

- While an NPN transistor conducts when a current flows into the base
- A PNP transistor will conduct when ...
  - No current flows into the base
- And stop conducting when ...
  - A current flows into the base.

PNP transistor as a switch (on)



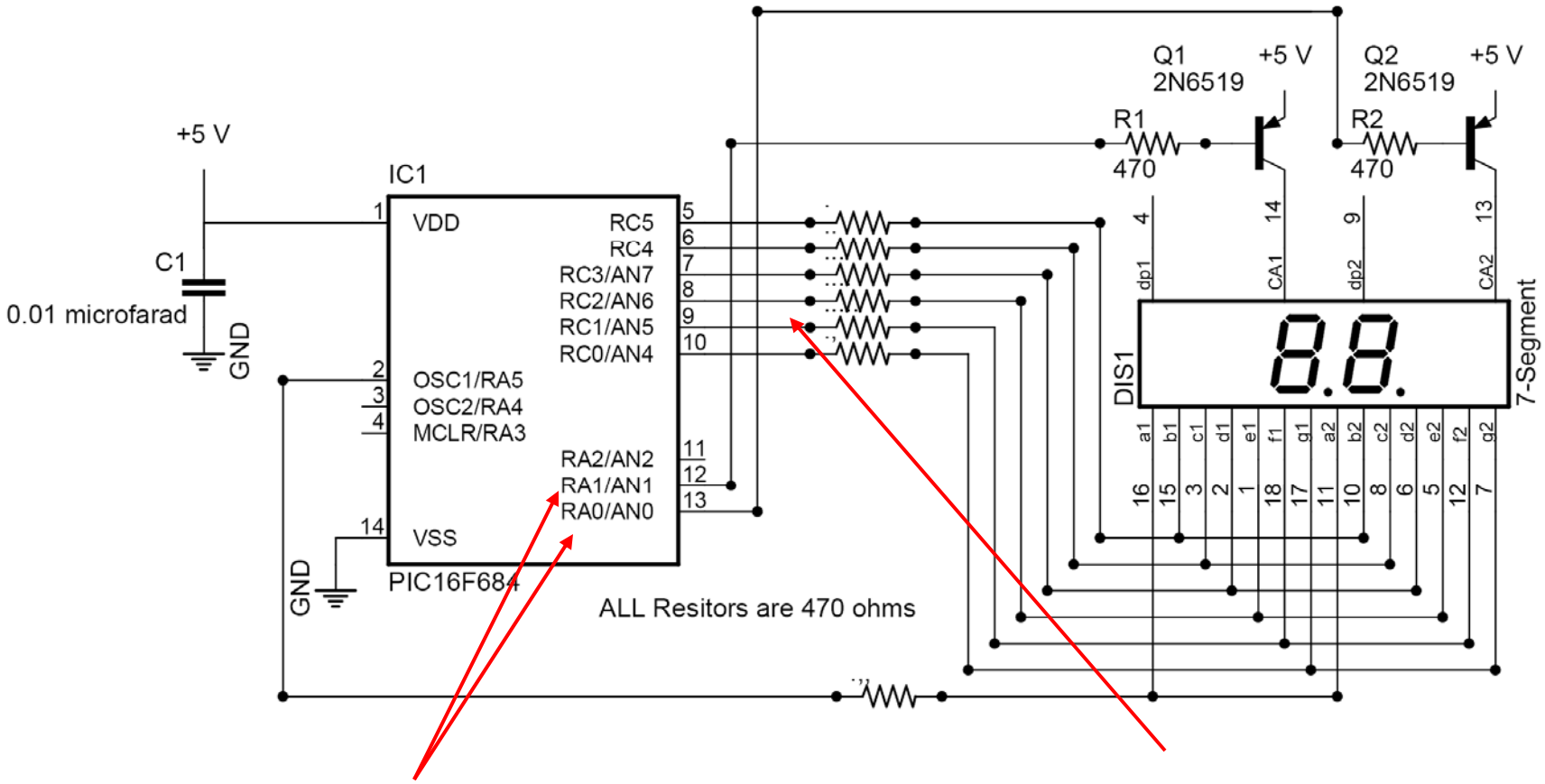
PNP transistor as a switch (off)



PNP Transistor – Our Labs

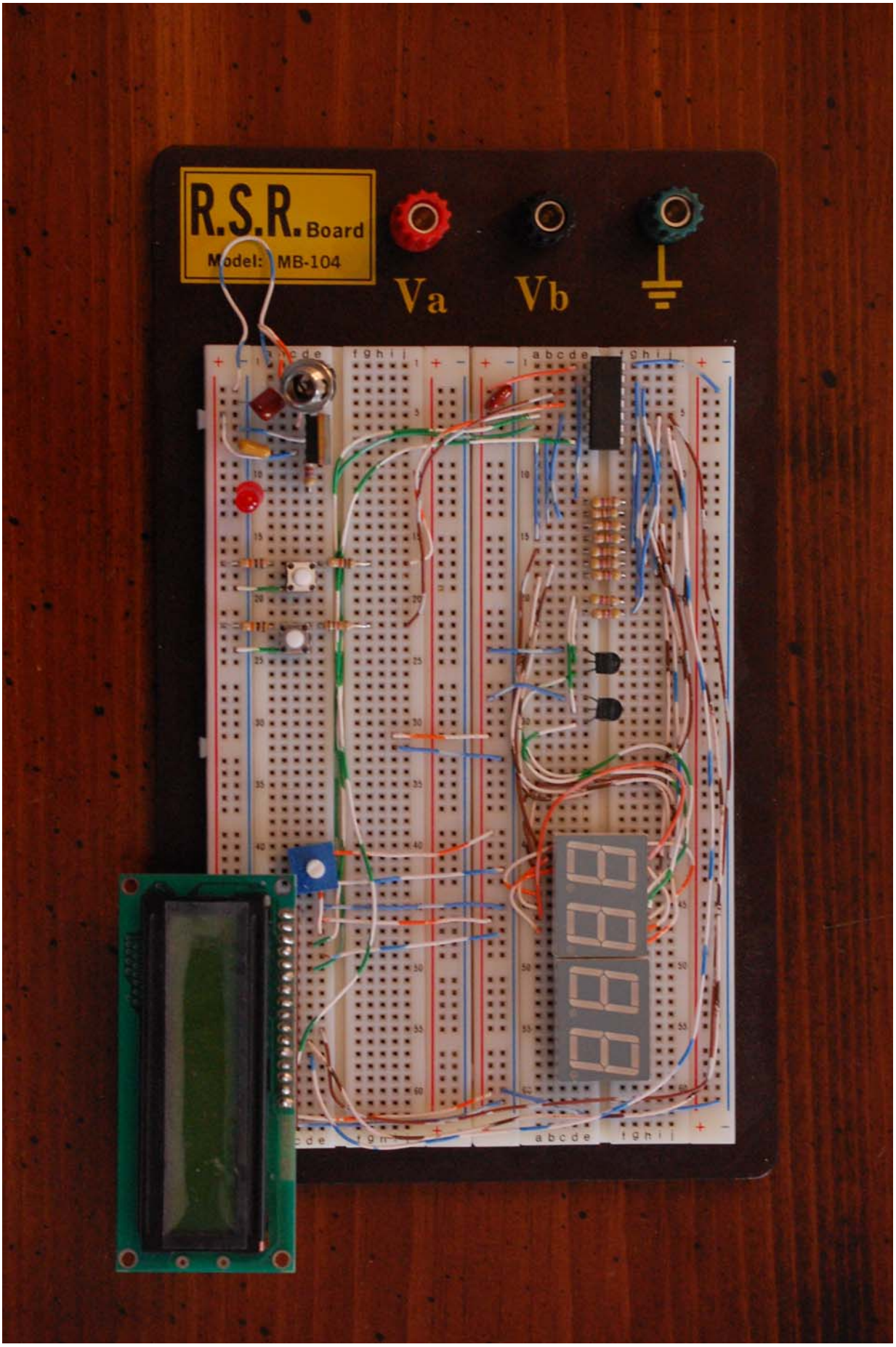
- We will be using the *PNP* Transistor ...
- Therefore ... the transistors we will use will conduct when ...
  - No current flows into the base
- And stop conducting when ...
  - A current flows into the base

Microcontroller and 7-Segment LED Circuit



0 = Selected
1 = Not Selected

0 = On
1 = OFF


R.S.R. Board
Model: MB-104
Va
Vb

Programming / Software

Programming

- Commands/instructions that we will encounter tonight
    - C commands
    - PIC16F684 control

“C” commands – Learned to Date …

C commands – Learned to Date

- C program structure
- Comments
- ***# include***
- Integer variables
- ***for***
- ***while***
- ***NOP()***
- ***functions***

C commands – Learned to Date (con't)

- ***if*** conditional
- ***else***
- ***else if***
- ***switch***
- ***Constant*** Declaration

C commands – Learned to Date (con't)

- ***Bitwise*** Operators

a == b	Tests for equality
a != b	Test for inequality
a > b	Greater than
a < b	Less than
a >= b	Greater than or equal to
a <= b	Less than or equal to

New “C” commands …

NEW C Commands

- ***Bitwise*** Operators
  - ***<<*** Shift Left
  - ***>>*** Shift Right
- ***arrays*** or ***look-up tables***

Bitwise Operators ...

Shifting Bits

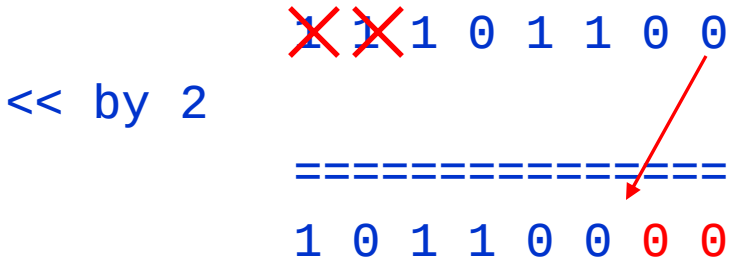
- Another type of useful bitwise operation is the shift
- We can shift bits either to the left (**<<**) or to the right (**>>**)
- Why use a shift? ...
  - To perform an operation on a certain bit
    - The bit of interest can be moved to a position where the operation can be performed

Binary Operators

- **<<** Shift Left
  - Drops the number of bits specified on the left
  - Replaced with zeros
- **>>** Shift Right
  - Drops the number specified on the right
  - Replaced with zeros

<< Shift Left

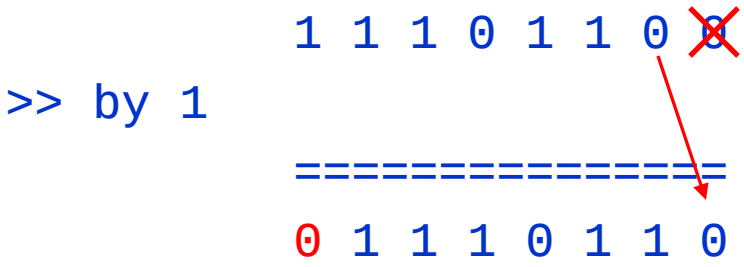
- Shift Left by 2



- We shifted the bit pattern to the left by 2 positions ... and ...
  - Replaced bits 0 and 1 positions with two zeros
  - The original bits 6 and 7 have been replaced by bits 4 and 5

>> Shift Right

- Shift Right by 1



- We shifted the bit pattern to the Right by 1 position ... and ...
  - Replaced in the bit 7 position with a zero
  - The original bit 0 has been replaced with bit 1

Applying the Shift Operator to the Microcontroller

```
PICKit1_LEDs = 0b010000;  // PORTA value to light D0
```

- We have the above variable in our program code
- We need to set RA4 equal to the value of the bit in position 4 (the 5th bit from the right ... recall we start with bit 0)
- What is one way of doing that?

Applying the Shift Operator to the Microcontroller

```
PICKit1_LEDs = 0b010000;  // PORTA value to light D0
RA4 = (PICKit1_LEDs & 0b010000) >> 4;
```

```
          0 1 0 0 0 0
AND (&)
          0 1 0 0 0 0
          ============
          0 1 0 0 0 0
```

- Shift Right by 4

```
          0 1 0 0 0 0
>> by 4
          ============
          0 0 0 0 0 1
```

- Resulting in RA4 equaling 1

Arrays or Look-up Tables ...

Arrays or Look-up Tables

- There are times when you want a variable to hold multiple values
- For example ... you may want the variable scores to hold 100 student test scores
- An ***array*** or ***look-up table*** is a data structure that can store multiple values of the same type

Declaring an Array

- The ***array*** format is as follows:

```
type array_name[size];
```

- To declare an array ... you must specify:
  - The array Type ... (int, char, float, etc)
  - Array name
  - Array size
    - To specify the array size ... you place the number of values that the array can store within brackets that follow the array name

Declaring an Array

- Example ...

```
int scores[100];
```

- The first array element is located at position zero … 0
- The last array element is located at [array size – 1]
  - Therefore … the first is at …

    scores[0];

  - The last element is at …

    score[99];

Initializing Arrays

- The following statement initializes the integer array scores to the values 80, 70, 90, 85, and 80

```
int scores[5] = {80, 70, 90, 85, 80};
```

- You do not need to fill all the space allocated ...

```
int scores[5] = {80, 70, 90};
```

- Depending on the compiler, it may assign 0 to the elements that you do not assign explicit values

Accessing Array Elements

- Using the previous example ...

```
int scores[5] = {80, 70, 90, 85, 80};
```

- If we had the following ...

```
i = 2;
scores[i];
```

- Scores would equal ...

```
scores[i] = 90
```

- Remember ... with 5 items ... they will be numbered 0 through 4

Alternate Method of Initializing Arrays

- Initializing arrays without specifying the size of the array ...
- If you had the following ...

```
int scores[] = {80, 70, 90, 85, 80};
```

- An array would be created to just fit the values specified ... 5

Utilizing Arrays With Microcontrollers

- We could create arrays to hold the PORTA and TRISA values for lighting D0 through D7 LEDs on the PICKit1

```
const char PORTA_Values[] = {
        0b010000,       // D0
        0b….            // D1
        …
        …
        0b000010};      // D7
```

- Then ... set PORTA as follows:

```
PORTA = PORTA_Values[0];        // Sets PORTA = 0b010000
```

General Programming ...

Software Thoughts ...

- Re-use your code where possible
  - Copy code that works from one program to another
  - Create yourself a “database”
- When writing Code ... do small pieces at a time
  - Once it’s working ... move to the next part
  - Integrate each part until the code is complete

Exam #1 Review ...

Review – Exam # 1

- Review all the lecture material
- Basic Microcontroller facts
- Examples of microcontroller applications
- How does the Microcontroller compare to the typical of a personal computer?
- Features and layout of the PIC16F684
- PICkit™ 1 Flash Starter Kit

Review – Exam # 1

- Microchip's MPLAB integrated development environment (IDE)
- PICC Pro C Compiler
- Typical Tasks for developing an embedded controller application
- Know the PIC16F684 Features
- Understand the Program and Data Memory Configurations

Review – Exam # 1

- PIC16F684 Interfacing
  - Switches
  - LEDs
  - Voltage Regulator Circuit
  - Seven-Segment LEDs

Review – Exam # 1 - C commands

- C program structure
- Comments
- ***# include***
- Integer variables
- ***for***
- ***while***
- ***NOP()***
- ***functions***

Review – Exam # 1 - C commands – (con't)

- ***if*** conditional
- ***else***
- ***else if***
- ***switch***
- ***Constant*** Declaration

Review – Exam # 1 - C commands – (con't)

- ***Tests***

a == b	Tests for equality
a != b	Test for inequality
a > b	Greater than
a < b	Less than
a >= b	Greater than or equal to
a <= b	Less than or equal to

Review – Exam # 1 - C commands – (con't)

- ***Bitwise*** Operators

  – ***<<*** Shift Left
  – ***>>*** Shift Right

- ***arrays*** or ***look-up tables***

Review – Exam # 1 – PIC16F684 Control

- PORTA
- TRISA
- PORTC
- TRISC
- R&#
- Configuration Word
- Numbering Formats

Material to Bring to the Exam ...

- Electronic versions of all your lab programs ... Labs 1 and 2
- PICkit™ 1 Flash Starter Kit ... including the PIC16F684 and USB cable
- A calculator ...
- Your laptop ... if you use it for programming the PIC16F684

ANY

QUESTIONS

RELATED TO

THE EXAM

????

Lab

A look at Lab #1 and Lab #2 Software …

Software You Used To Control The PICkit #1

- Lets look at how you went about developing your software ...

Next Class ...

Next Class Topics

- **Exam #1**

HomeWork

Homework

1. Study for Exam #1

2. Bring required ***Exam Materials*** to class next week

3. Read as required the C commands discussed in today's lecture found in the text (Programming in C)

4. Read the PIC16F684 data manual sections for the registers encountered tonight

5. Finish Lab #2

6. Lab report for Lab # 2 is due October 19, 2010

Time To ...

Start the Lab ...

References

1. PIC16F684 Data Sheet 41202F