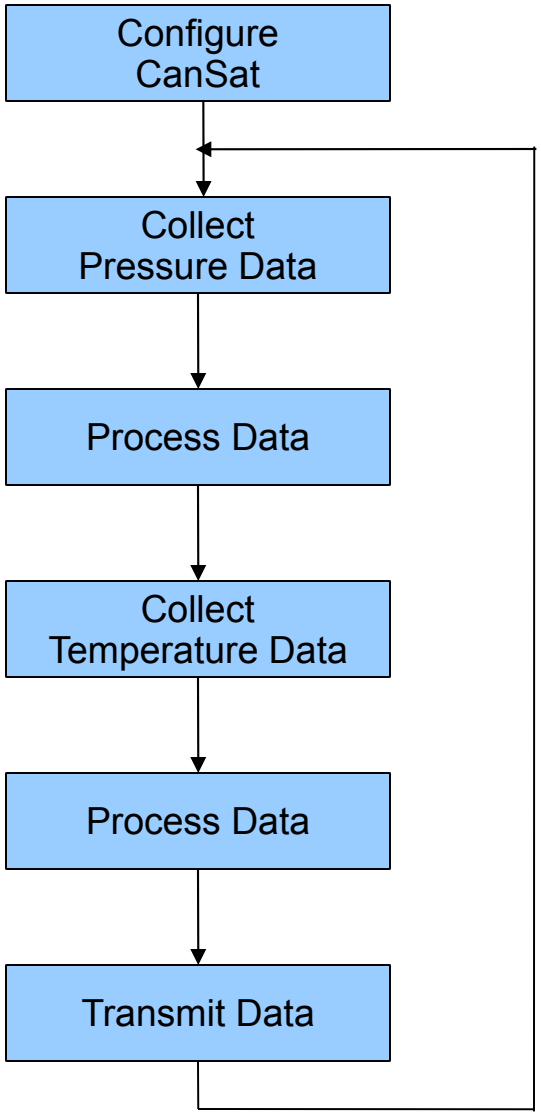
Software Development

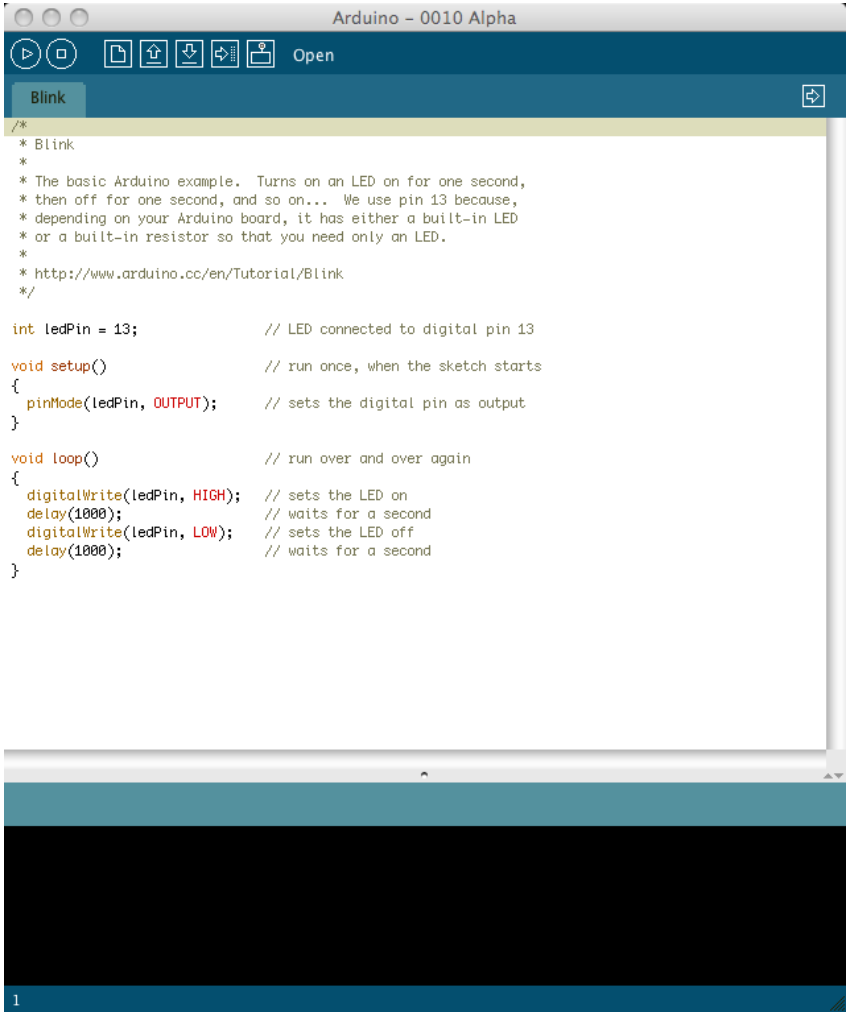
CanSat Software Development

- In order for the CanSat to work, software needs to be developed for the data handling unit. The software needs to collect sensor data, process the data, and send it to the ground station.
- This section will teach the basics of the programming language to allow completion of the software.
- The drawing to the right is called a flow chart. The flow chart shows what the software is to do. What is shown is the software continuously loops.



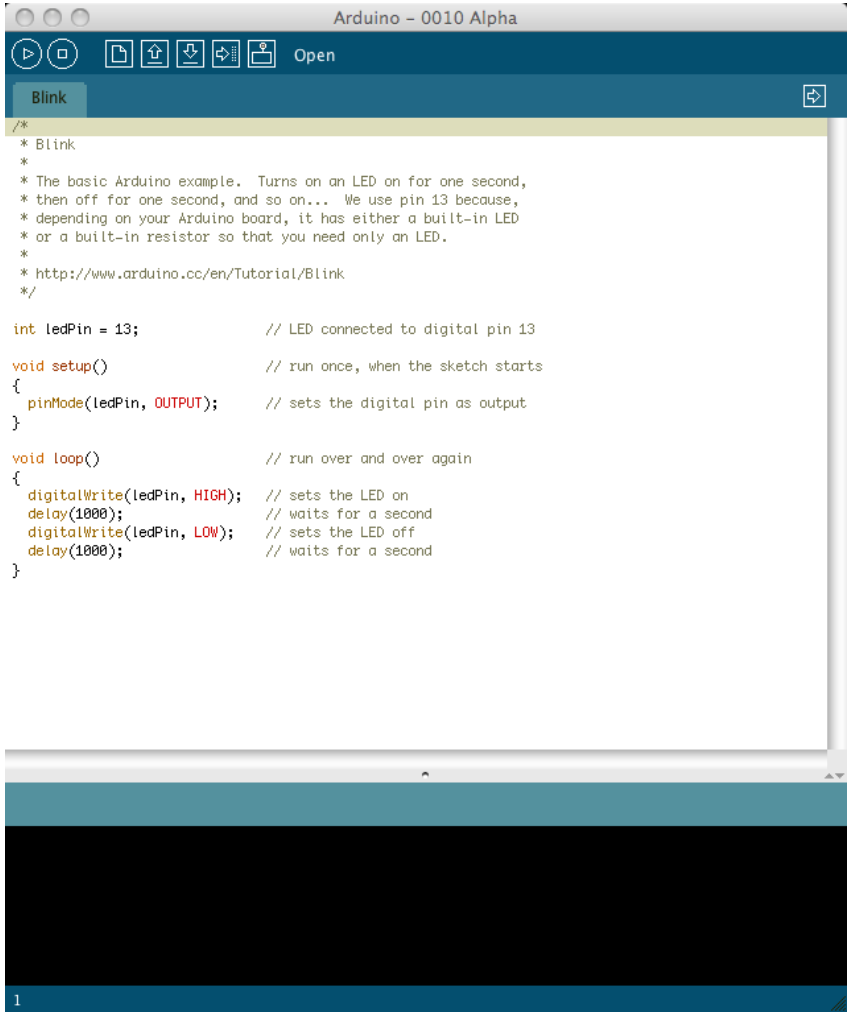
Software Development Tools

- The Data Handling Unit or DHU will be programmed in the C language
  - To upload the program to the DHU requires a programmer adapter
  - The development software is called Arduino
    - Arduino provides a method to write software, upload it to the DHU and test the software
    - For more info on Arduino, go to www.arduino.cc



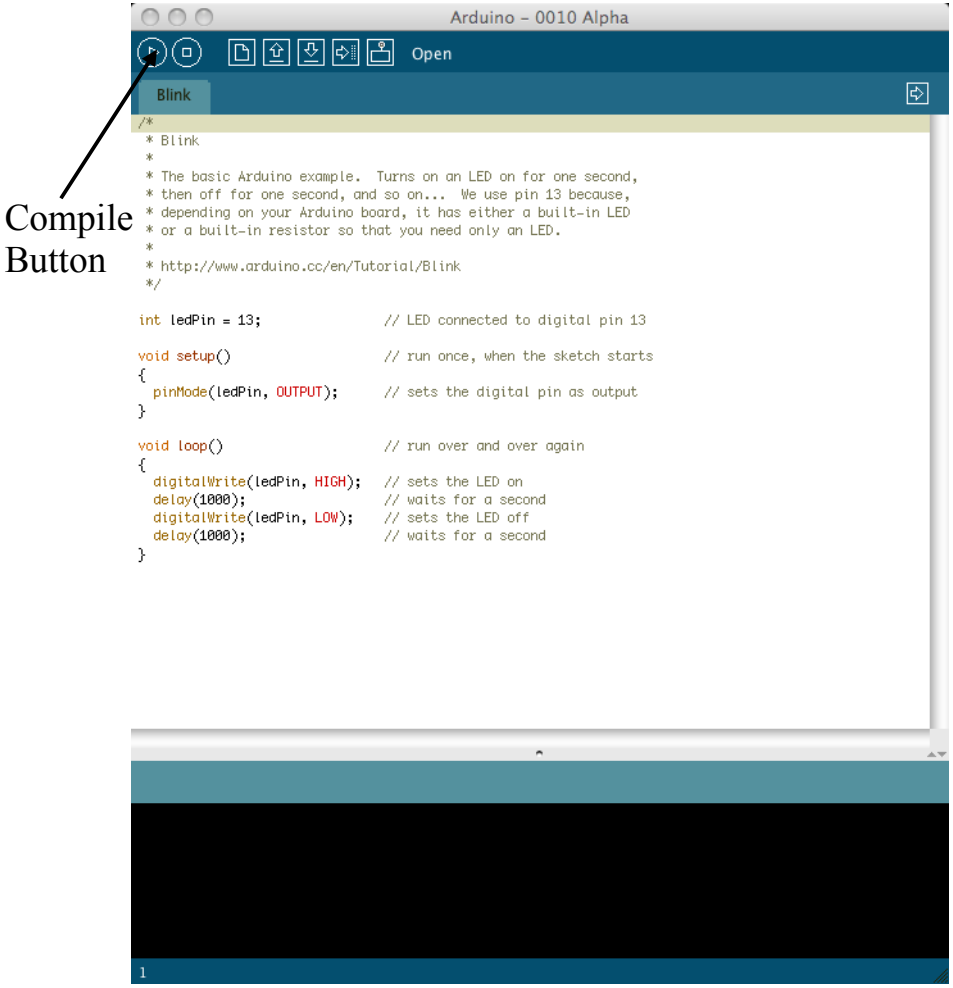
Setting up the Compiler

- Installing the software is simple. Drag the Arduino-0018 folder to the C drive. That is it.
- On the C drive, open the Arduino folder.
- Double Click on the Run Icon.
- The software will start.
- Setting up the compiler is simple
  - Click on the “Tools” menu
  - Select “Board”
  - Select “Arduino Diecimila”
  - Click on the “Tools” menu
  - Select “Serial Port”
  - Click on the appropriate COM port.
- All Done



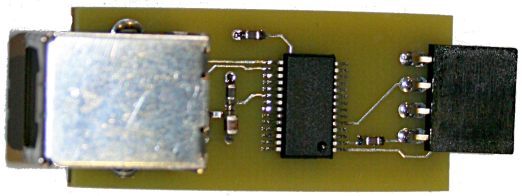
Writing a Program

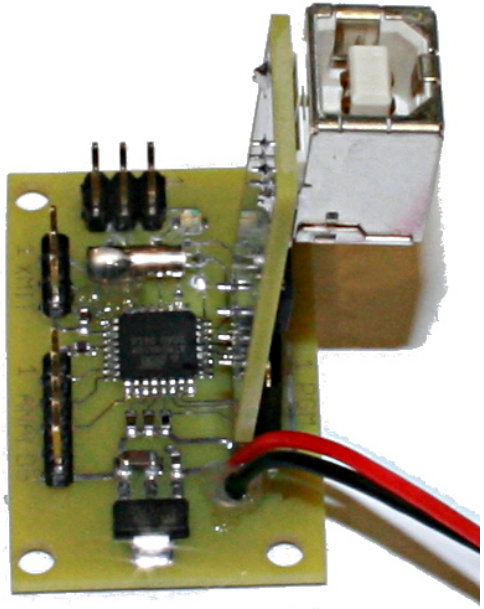
- The Development Software Includes a Program Editor.
- Start the Program.
- Select the Menu 'File' and Select 'New'.
- The editor window will clear.
- Now it is time to write a program.
- You can also save the file before compiling. Create a directory to save the files.
- To compile, click on the arrow button at the top.



Programming the Data Handling Unit

- A programmer module is needed to upload software to the Data Handling Unit (DHU). One end plugs into the PC's USB port. The other end is a 4 pin header connector that plugs into the DHU.
- Connect the 4-Pin Connector to the CanSat Electronics Board as shown in the picture.
- Plug in the 9 Volt Battery: Make Sure You Don't Try Connecting the Battery in Backwards





Writing the First Program

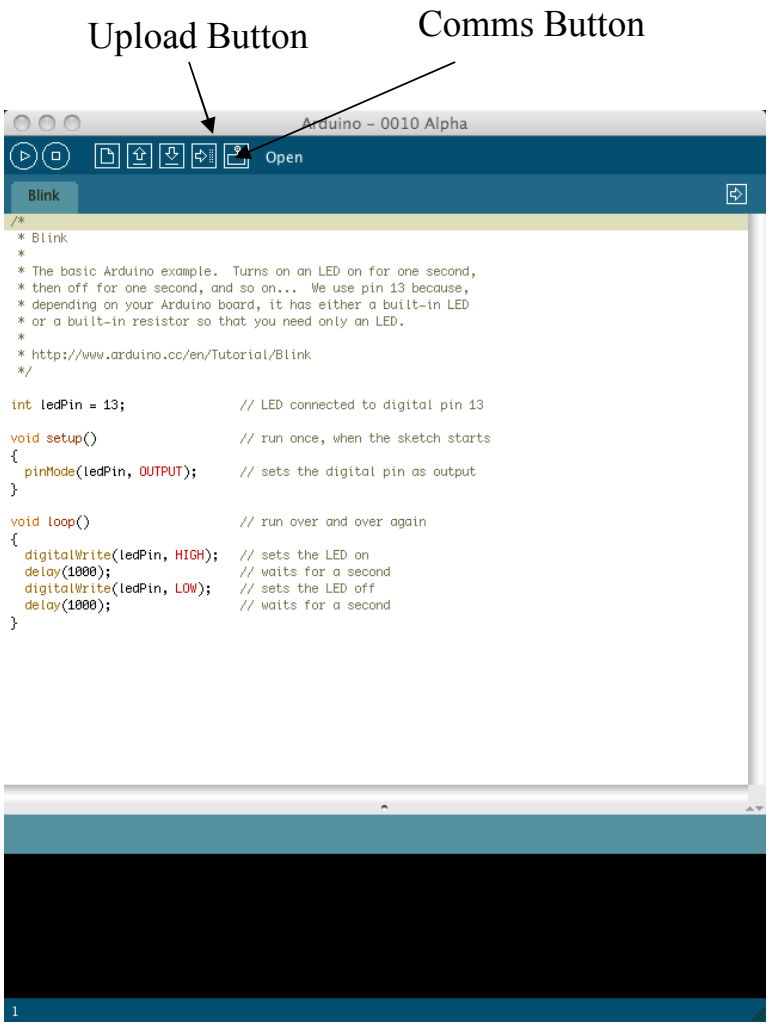
- The First Program Will Print Text to the Computer
- In the Editor, Enter the Following:

```
void setup()
{
   Serial.begin(38400);
}
void loop()
{
   Serial.println(“Hello world”);
}
```

- Once Entered, click the compile button
- If an Error Is Indicated in the Bottom Status Window, Look in the Editor Window for Any Mistyping

Uploading to the DHU

- Click on the upload button.
  - This will upload the software to the DHU.
- Click on the Comms button.
  - You will see a Selection at the bottom left for selecting the data rate. Set it to 38400.
  - Anything being printed will show up in the bottom window



Understanding Bits

- To Understand Variables Which Will Be Discussed Soon, Bits Need to Be Understood; Computers Use Bits in Many Ways
  - A Bit Is a Single Piece of Information That Can Be at Two States, a Logic Level 1 and a Logic Level 0
  - The State Is Called a Binary State Which Means a Bit Can Only Be Any of Two States
  - Not Much Can Be Done With a Single Bit; To Be Useful, the Computer Can Group a Number of Bits Together to Represent Numbers or Other Information, There Are Standard Groupings:
    - Nibble Is Four Bits Together
    - Byte Is Eight Bits Together
    - Word Is 16 Bits Together
    - Long Is 32 Bits Together

Byte

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

- There Is an Order to the Bits and a Weighting Scheme So That Numbers Can Be Represented
  - Let's Take a Byte - It Has Eight Bits
    - As Shown, the Least Significant Bit Has a Weight of Zero
    - To Calculate the Value, Calculate 2 to the Power of the Digit Which Is Zero
    - The Result Is One
  - Take the Next Bit
    - It Has a Weight of 1. 2^1 Is Equal to 2
    - The Next Bit Is 2^2 Which Is Equal to 4; 2^7 Equals 128
  - To Figure Out the Value of the Byte, Add up All the Weights for the Bits That Equal 1
- For Example, the Byte “10111001” Equals 185; $2^7 + 2^5 + 2^4 + 2^3 + 2^0 = 185$

Using Variables

- Now, It Is Time to Learn About Variables
  - Variables Are Used to Store Temporary Information in Memory During Program Execution
  - There Are Five Different Types of Variables; Each Is Used to Contain a Certain Size Value
    - char Defines an Eight Bit Variable That Can Have a Range From -127 to 127
    - Int Defines a 16-bit Variable That Can Have a Range From -32767 to 32767
    - Long Defines a 32-bit Variable That Can Have a Range From -2147483647 to 2147483647
- Variables Are Declared With the First Character Being a Letter

```
char dog; // this declares variable dog to be a byte.
int t34;  // this declares variable t34 to be a word.
```

How to Print Variables

- There are two functions available to print information
- The first program showed how to print text.
- To Print Variables, Enter the Following:

```
void setup()
{
   Serial.begin(38400);
}
void loop()
{
   int a;
   a = 78
   Serial.print(a);
}
```

- The Above Prints the Value of Variable 'a' As a Decimal Number

Printing Multiple Variables

- Printing Multiple Variables requires the use of multiple print commands.
  - There are two different print commands
    - Serial.print() will print on a line. Any other print commands will continue on the same line.
    - Serial.println() will print on a line and cause any other prints to start on the next line.

```
void loop()
{
  char a,b;
a = 5;
b = 8;
Serial.print(a);
   Serial.println(b);
}
```

Control the Light

- Now It Is Time to Learn How to Use the Digital Control Ports
  - A Light Emmiting Diode or LED Is Included on the CanSat Electronics Board; It Is a Semiconductor Device That Emits Light When Electricity Flows Through It. It is connected to port 8.
- First, the digital port needs to be configued in the setup() function
  - pinMode(8,OUTPUT);
- To Turn the LED Off, Use the Command:

```
writeDigital(8, HIGH);
```

- Turn Turn the LED on, Use the Command:

```
writeDigital(8,LOW);
```

- That Is All There Is to Controlling the Digital Ports
- When You Set the Port to High, You Are Setting the Port Signal to a High Level Which Is a Logic Level 1 Which Is 5 Volts
- When You Set the Port Low, You Are Setting the Port Signal to a Low Level Which Is a Logic Level 0 Which Is 0 Volts

More Light Control

- Now, Let's Write a Program That Will Continuously Flash the LED on and off
  - To Do This, the Program Has to Run in a Loop; Loops Are Easy to Set up
  - First a Label Is Used to Indicate the Start Location of the Loop
  - Then Later in the Program Is the Goto Statement That Specifies the Label for the Program to Go to
  - Enter and Run the Following Example:

```
void setup() {
  pinMode(8,OUTPUT);
}

void loop() {
digitalWrite(8,LOW);
digitalWrite(8,HIGH);
}
```

- The Program Will Flash the LED As Fast As the Processor Can Execute the Program
  - You Probably Cannot See the Flashing
  - That Can Be Fixed Next

Adding Delay

- There Is a Function Called delay();
  - The delay() Function Tells the Processor to Stop Operating for a Period of Time
  - The delay() Function Will Stop the Processor in One Millisecond Increments; That Is 1/1000th of a Second
  - Insert the Pause Command in the Program As Shown:

```
void loop()
{
  digitalWrite(8,LOW);
  delay(1000);
  digitalWrite(8,HIGH);
  delay(1000);
}
```

Delays

- The Program Should Have Turned the LED on for One Second and Turned It off for One Second
- Experiment and Change the Pause Time to Different Values

Doing Math

- Performing Mathematical Functions Is Pretty Simple; It Is Like Writing Equations
  - Try the Following in a New Program in the loop() function:

```
int a;

int b;

int c;

a = 5;

b = 6;

c = a + b;

Serial.print(“The answer is: ”);

Serial.println(c);
```

Floating Point Math

- Up to Now, Any Numbers and Math Were Based on Integer Numbers Which Are Whole Numbers
  - Floating Point Numbers Allows the Processor to Work With Fractional Numbers

- The Floating Point Format Uses 32-Bit Variables
  - To Use Floating Point, Declare the Variables As “Float”
  - The Followng Example Shows How It Works:

```
void loop()
{
  float Temp;
  Temp = 1.543;
  Serial.println(Temp,3);
}
```

How To Print Floating Point Numbers

- The number after the variable specifies how many decimal places are displayed.
- Write a program to display a floating point number and change the number of decimal places and see what happens.

Floating Point Math

- Writing Math Operations Is the Same As Integer Math
  - To Add, Use +
  - To Subtract, Use -
  - To Multiply, Use *
  - To Divide, Use /

- Below Are Examples of How to Use the Math Functions. Enter the Program on the Next Page

```
float Temp1;
float Temp2;
Temp1 = 3.4156;
Temp2 = temp1 * 2.345;   // multiply temp1 with 2.345
Temp2 = temp1 + 3.234;   // add temp1 and 3.234
Temp2 = temp1 / 5.0;     // divide temp1 by 5.0
Temp2 = temp1 - 34.0;    // subtract 34.0 from temp1
```

Summary

- In This Section, You Should Be Able to Write Simple Programs; You Should Have an Understanding of:
  - How to Declare and Use Variables
  - Write Program Loops
  - Do Math Operations on Variables
  - Do Floating Point Operations on Variables