TO DO

- Continue to bring your laptop and power cord to class for the rest of the quarter.
- Finish constructing your Hapkit (with new laser-cut sector pulley) before performing the lab.
- Make yourself a shelf label for your Hapkit and accessories (there are sharpies, post-it notes, and tape in the near the shelves). Two people per shelf, please.
- Remember to do the online videos and quizzes for this week by tomorrow. The programming portion is optional -- there are no quizzes in that section.
- Get checked off on Lab 4 before you leave today!

Introduction to the Hapkit board

Allison M. Okamura
Stanford University

Programming Hapkit

Arduino is a single-board microcontroller that makes using electronics in multidisciplinary projects relatively easy

The Hapkit board is based on the Arduino design, with some added features

The Hapkit board can be programmed using the same Arduino integrated development environment (IDE) as an Arduino board

Follow the instructions in the handout to download, install, and test the Arduino software

Example Hapkit Program

Code:

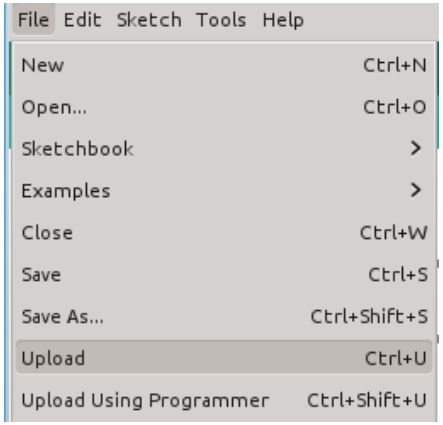
```
// Blink the pin 13 LED
void setup() {
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);
}
```

The Arduino programming language is similar to C and java

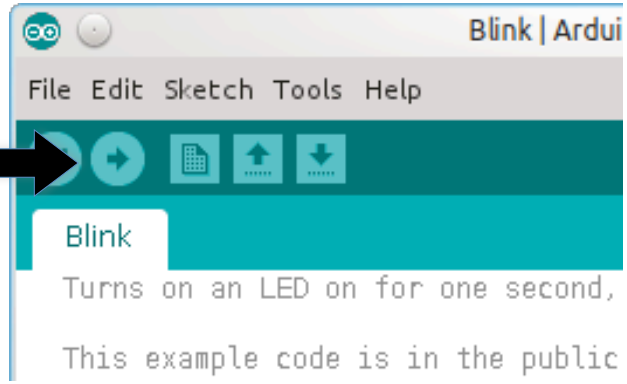
material courtesy Paulo Blikstein

Running a Hapkit Program



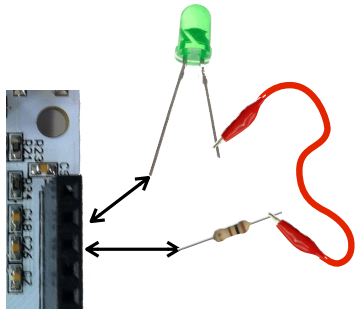
Click
File > Upload

or click the Upload button



Clicking Upload sends the code from the computer to the Hapkit board

Your code is now running on the Hapkit Board!
What is the LED doing?



material courtesy Paulo Blikstein

Understanding the code

```
// Blink the pin 13 LED
void setup() {
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);
}
```

void setup() { … }
Any code inside the setup function runs *once* to setup the Arduino

void loop() { … }
Any code inside the loop function loops over and over again

material courtesy Paulo Blikstein

The Serial Monitor

- The Arduino environment's built-in serial monitor can be used to communicate with your Hapkit board
- In the menu, click on **Tools > Serial Monitor** to view
- You can use the following code to interact with the serial monitor:

```
void setup() {
  Serial.begin(9600);      // open the serial port at 9600 bps
}
```

to print to the serial monitor:

```
Serial.println("Hello World!");

int my_variable = 0; // variable
Serial.println(my_variable);
```

to read from the serial monitor:

```
if (Serial.available() > 0)
{
  char inByte = Serial.read();
}
```

Programming syntax/hints

- Use // before any text you want to have as a comment (and comment well!)
- Each statement must end with a ; (semicolon)
- You must declare variables before you use them
- Call built-in Arduino functions to perform I/O (input/output)
- See http://arduino.cc/en/Reference/HomePage for a language reference that describes structure, variables, and functions
- If you have never programmed before, the easiest way to learn is by looking at and modifying example programs. Don't modify too many things at once before testing your code. Many examples under File > Examples

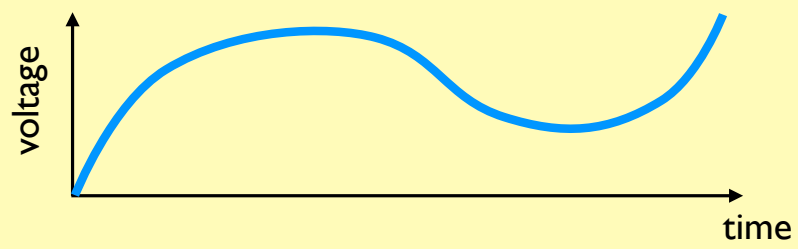
Hapkit Board
Analog Inputs

Allison M. Okamura
Stanford University

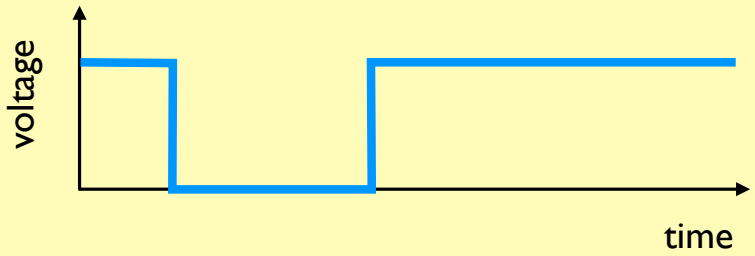
Digital vs. Analog

In mechatronic systems, both **analog** (continuous) and **digital** (discrete) signals are used to transmit information

Analog information varies continuously in amplitude



Digital information is binary (HIGH=1 or LOW=0)



Binary counting

Because digital systems compute in terms of 1's and 0's,
we need to understand the binary, or base 2, counting system

Binary	Decimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
1000	8
...	...
1111	15

- Group exercise: Count to decimal 10 using finger binary on one hand
- The binary numbers in the table on the left have 4 bits

Q1: How many numbers can be represented by 4 bits?

- n bits can represent 2^n decimal numbers
- 8 bits is a byte (this is a common unit)

Q2: How many numbers can be represented by a byte?

Analog-to-Digital (A/D) Conversion

When we measure voltage from a sensor (like your MR sensor), the Hapkit Board uses an A/D conversion

Process:

- A sensor generates an analog voltage between 0 and 5 V on a pin
- The A/D converter reads this voltage and converts it to a decimal number between 0 and 1023
- You use the function analogRead(pin#) in your code to read this decimal number, based on the voltage at the specified pin.

Q1: How many bits of resolution does this A/D converter have?

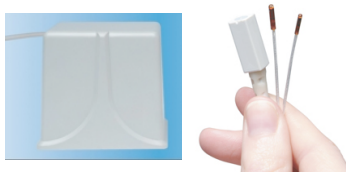
Q2: If the sensor outputs 2.5 V, what will the A/D converter output?

Hapkit position sensing

Allison M. Okamura
Stanford University

Sensor types

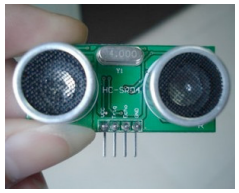
- magnetic
- optical
- acoustic
- inertial
- **mechanical**
  (our focus, since these are the sensors typically integrated with the actuator in kinesthetic haptic devices)



magnetic: TrakStar, Ascension

optical: Polaris, NDI

optical: Microsoft Kinect



acoustic: ultrasonic proximity sensor, BiF

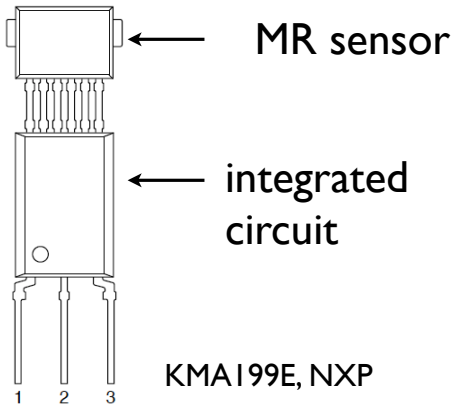
inertial: wearable IMU, MotionNode

mechanical: Faro arm

Mechanical trackers

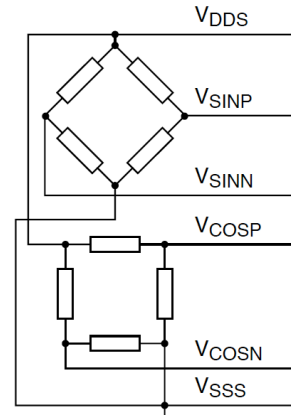
- Ground-based linkages most commonly used
- Joint position sensors
  - digital: optical encoders are most common
  - analog: magnetic sensors and potentiometers are most common

Magnetoresistive (MR) angle sensors



KMA199E, NXP

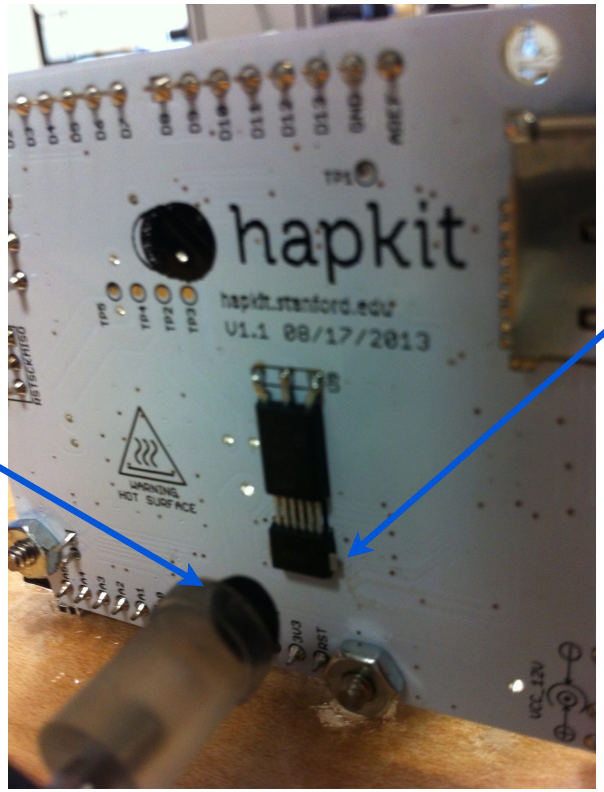
- magnetoresistive materials change their electrical resistance when an external magnetic field is applied
- the resistance depends on the angle between the magnetization vector of the ferromagnetic material and the direction of current flow (resistance is largest if they are parallel)
- often 4 sensors are connected in a Wheatstone bridge configuration (similar to strain gages)



KMA199E, NXP

MR sensor on the Hapkit board

magnet
(rotates with motor)



MR sensor

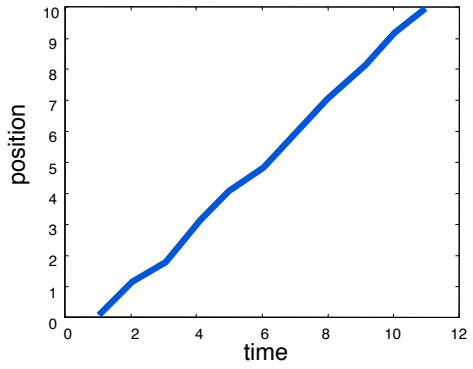
Measuring velocity

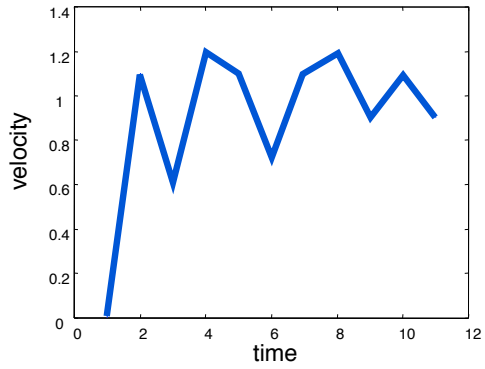
- Differentiate position
  - advantage: use same sensor as position sensor
  - disadvantage: get noisy signal

Discrete differentiation

- Many different methods
- Simple example:
  - average 20 readings = P1
  - average next 20 readings = P2
  - where t is the the period of the haptic loop
- Differentiation increases noise

$$V = \frac{P1 - P2}{t}$$

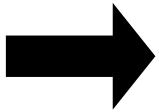




Computing haptic interaction point velocity

- Recall that the forward kinematics tells us the haptic interaction point position based on the joint position and geometry of the device
- How do we calculate velocity?
- For a one-degree-of-freedom system, the same transmission ratio used to compute the position can be used for the velocity.

$$x_{\text{handle}} = \frac{r_{\text{handle}} r_{\text{pulley}}}{r_{\text{sector}}} \theta_{\text{pulley}}$$



differentiate

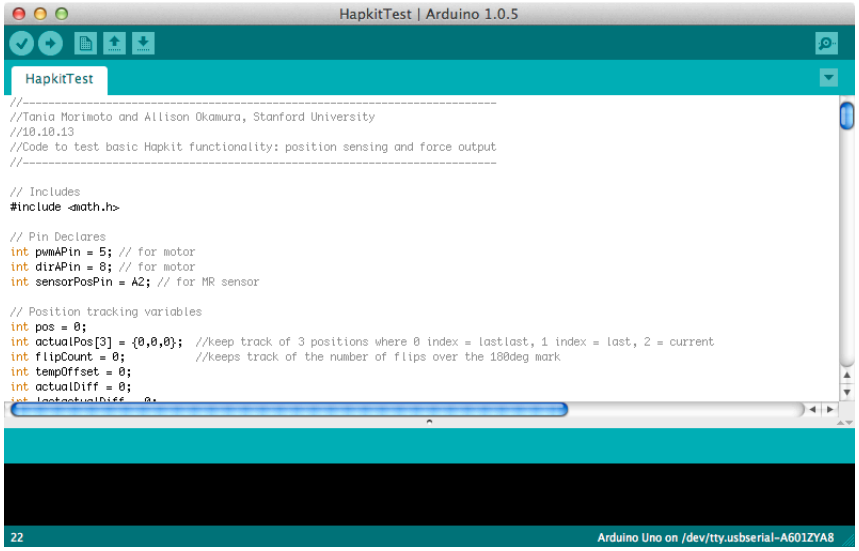
$$v_{\text{handle}} = \frac{r_{\text{handle}} r_{\text{pulley}}}{r_{\text{sector}}} \omega_{\text{pulley}}$$

Walk through of Hapkit test sketch

Allison M. Okamura
Stanford University

Get the sample code from the online course website:

https://class.stanford.edu/courses/Engineering/ME-20N/Fall2014/
> Courseware > Week 4: Hapkit Mechatronics I
> Material presented/distributed in class
> Download zip file of Arduino code.



Hapkit position sensor calibration

Allison M. Okamura
Stanford University

Activity: Calibrate Sensor

We need to calibrate the Hapkit so that we know what positions and forces we are working with in real, translational units (meters and Newtons)

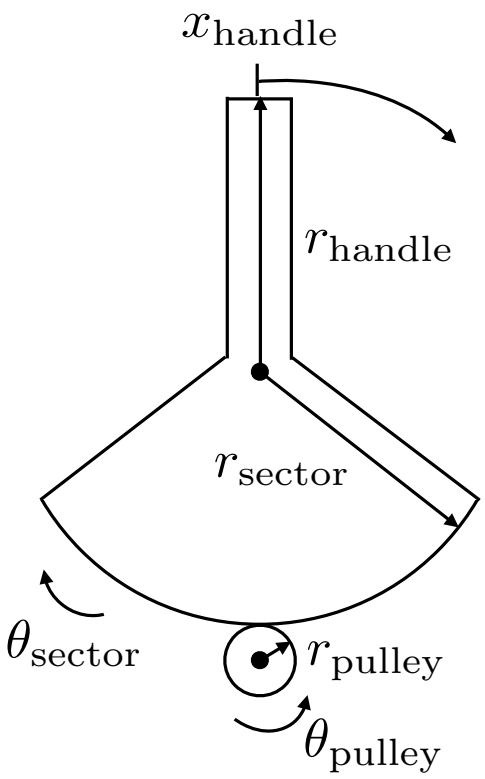
MR sensor calibration: Use angle markings on your Hapkit sector, the Serial Monitor (SM), and r_{handle} to determine x_{handle}.

1. ***Manually record*** a series of about 10 sector angles (degrees) and their corresponding A/D output values.
2. ***Plot these in Excel*** with angles on vertical axis (y) and A/D outputs on horizontal axis (x) and fit a straight line.
3. ***Record*** the m and b ***parameters*** of the fit (y = mx + b).
4. Convert sector angle to radians, and then ***use kinematics*** to convert sector angle to ***handle position in meters***.
5. ***Add the equations*** from steps 3 and 4 ***to your code***, and test using SM.

Have your calibration checked off by Allison before you leave today!

(I will stay for office hours today until 4 pm, so I can also check off during office hours)

Reminder: Hapkit Kinematics



$$r_{\text{pulley}}\theta_{\text{pulley}} = r_{\text{sector}}\theta_{\text{sector}}$$

$$x_{\text{handle}} = r_{\text{handle}}\theta_{\text{sector}}$$

$$x_{\text{handle}} = \frac{r_{\text{handle}}r_{\text{pulley}}}{r_{\text{sector}}}\theta_{\text{pulley}}$$