UNIVERSITY OF TWENTE.

LEGO MINDSTORMS & ARDUINO

PRACTICAL SESSION 2

Part of SmartProducts

LEGO MINDSTORMS & ARDUINO

PRACTICAL SESSION 2

Fjodor van Slooten
W241 *(Horst-wing West)*
f.vanslooten@utwente.nl

- Arduino programming basics
- Driving robot cars
- Lego sensors advanced
- Assignment

Additional Online Materials

- Arduino tutorials
- Java tutorials
- UI Prototyping
- Evshield

Please store kits properly.
Stack max. 4 pieces

slides @ vanslooten.com/appdev

LAST FRIDAY

A bit chaotic, two issues:

- Problem with Dabble library
- Motors spinning out of control (when steering)

Challenge … for some of you difficult

Most groups managed to get it working

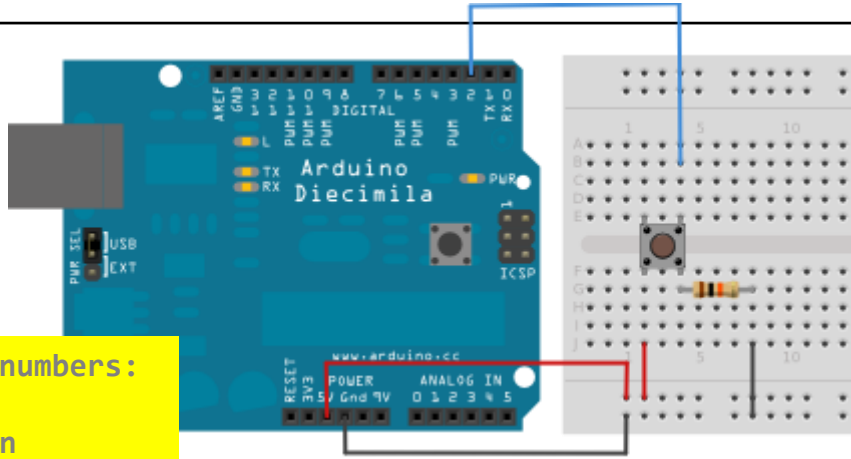
We all learned…

Changes today:
More choice: choose what you think is useful for project!
Options vary in difficulty, *try to do as much as you can*

If your kit is missing pieces/materials, you can get replacements from teacher!

ARDUINO PROGRAMMING BASICS

File > Examples > 02.Digital > Button

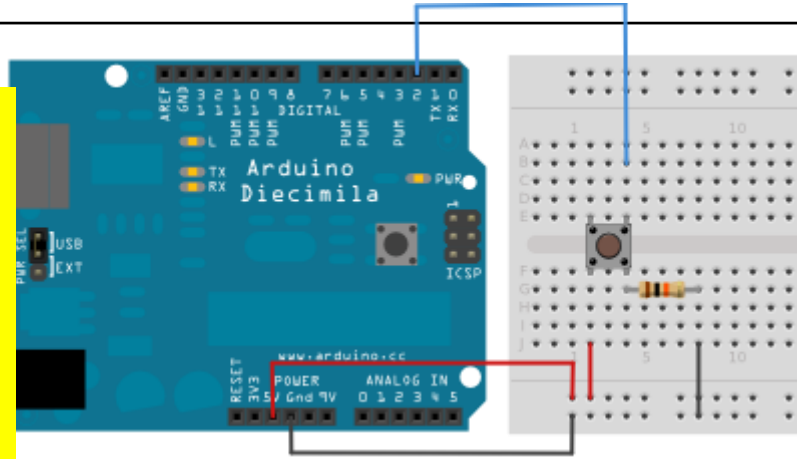



```
// constants won't change. They're used here to set pin numbers:
const int buttonPin = 2;     // number of pushbutton pin
const int ledPin =  13;      // number of onboard LED pin

// variables will change:
int buttonState = 0; // variable for reading pushbutton status

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}
```

arduino.cc/en/Tutorial/Button

ARDUINO PROGRAMMING BASICS

```
void loop() {
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed.
  // If it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  } else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```



Does not work as expected...?
Check out next example: 'debounce'

arduino.cc/en/Tutorial/Button

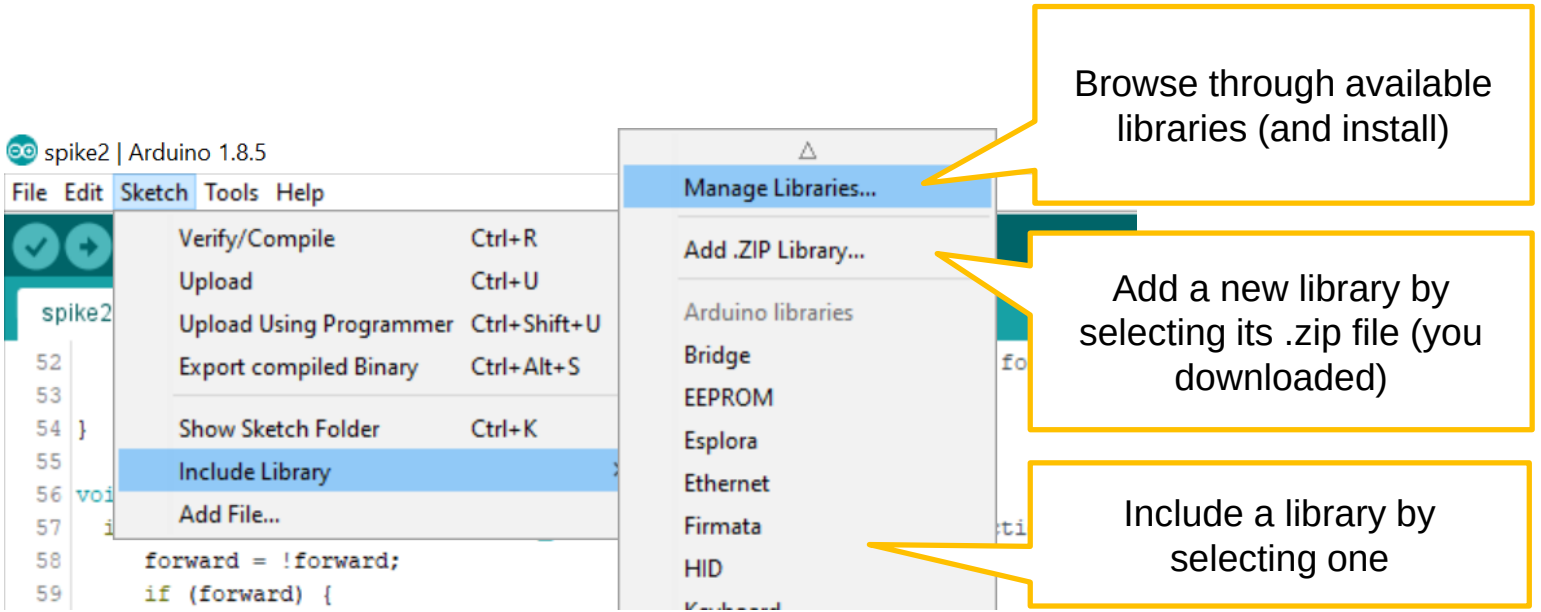
arduino.cc/en/Tutorial/Debounce

USING LIBRARIES

MAKE PROGRAMMING EASIER

arduino.cc/en/Main/Libraries

- Libraries extend functionality
- `Documents\Arduino\libraries` contains folders with libraries



ARDUINO MEMORY

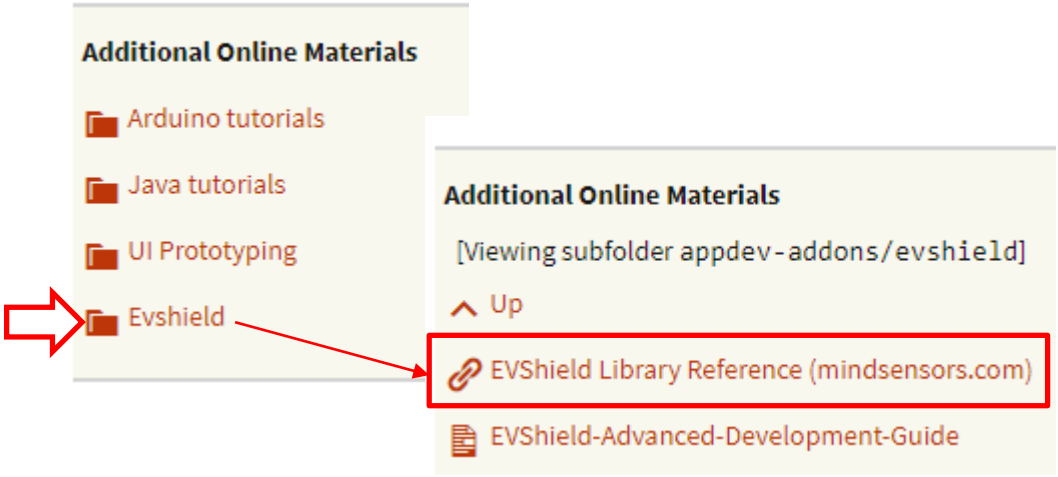
- Libraries are nice, as they allow easy programming
- But libraries usually use a lot of memory
- Combining multiple (large) libraries might result in to much memory usage

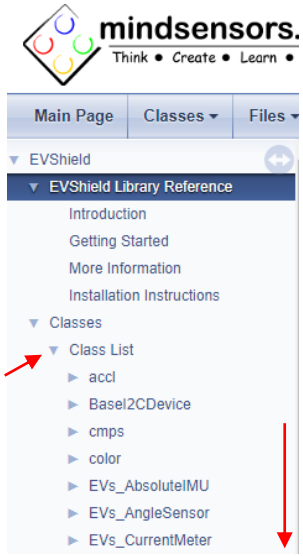
Example: combination of EVShield library and Blynk 'does not fit' together in memory of Arduino

learn.adafruit.com/memories-of-an-arduino/optimizing-sram

ACCESS EVSHIELD LIBRARY REFERENCE

Motor… commands?





scroll down until "**EVShieldBank**", click that

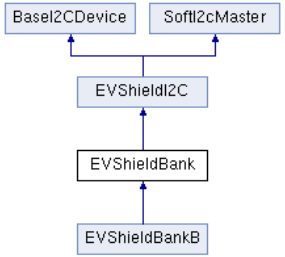
This is also available on your own computer:
Documents\Arduino\libraries\EVShield\html\ index.html

EVShieldBank Class Reference

This class defines methods for the **EVShield** Bank(s). More...

```
#include <EVShield.h>
```

Inheritance diagram for EVShieldBank:



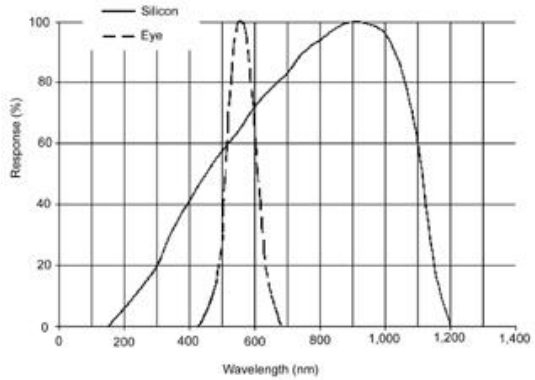
Public Member Functions

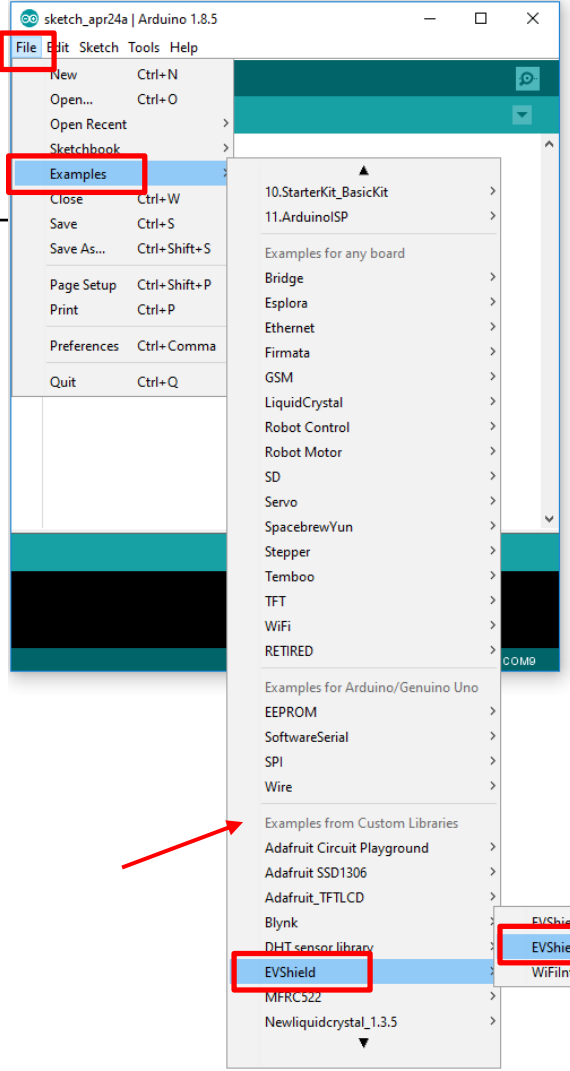
	EVShieldBank (uint8_t i2c_address=**SH_Bank_A**)
int	**evshieldGetBatteryVoltage** ()
int	**nxshieldGetBatteryVoltage** ()
uint8_t	**EVShieldIssueCommand** (char command)
bool	**motorSetEncoderTarget** (**SH_Motor** which_motor, long target)
long	**motorGetEncoderTarget** (**SH_Motor** which_motor)
bool	**motorSetSpeed** (**SH_Motor** which_motor, int speed)
int8_t	**motorGetSpeed** (**SH_Motor** which_motor)
bool	**motorSetTimeToRun** (**SH_Motor** which_motor, int seconds)
uint8_t	**motorGetTimeToRun** (**SH_Motor** which_motor)

Methods of **EVSHieldBank** Class, a lot motor-related

EXAMPLES

NXT LIGHT SENSOR





DIFFERENTIAL DRIVE

USE DRIVING WHEELS FOR ADDITIONAL STEERING

Used in **rover_bt_dabble.ino**

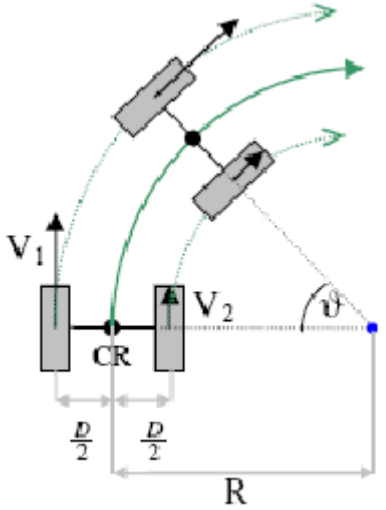
```
void differentialDrive(SH_Direction dir) {
  if (evshield.bank_b.motorGetEncoderPosition( SH_Motor_1 ) > -3 && evshield.bank_b.motorGetEncoderPosition( SH_Motor_1 ) < 3) {
    // just drive both motors equally
    evshield.bank_a.motorRunUnlimited( SH_Motor_Both, dir, speed);
  } else {
    float steer_pos = evshield.bank_b.motorGetEncoderPosition( SH_Motor_1 ) / 57.296; // calculates current steering position in radians
    float steer_radius = car_wheelbase * tan(1.571 - steer_pos); // calculates the radius, from centerline of car to center of steercircle
    float ratio_L = (steer_radius - (car_rear_track / 2)) / steer_radius;
    float ratio_R = (steer_radius + (car_rear_track / 2)) / steer_radius;

    evshield.bank_a.motorRunUnlimited( SH_Motor_1, dir, speed * ratio_L);
    evshield.bank_a.motorRunUnlimited( SH_Motor_2, dir, speed * ratio_R);
  }
}
```

Driving straight?

Yes, so drive motors equally

Turning, so calculate ratio

We use a ratio here: when turning, the inner wheel rotates slower, the outer faster



ICR (ICC)

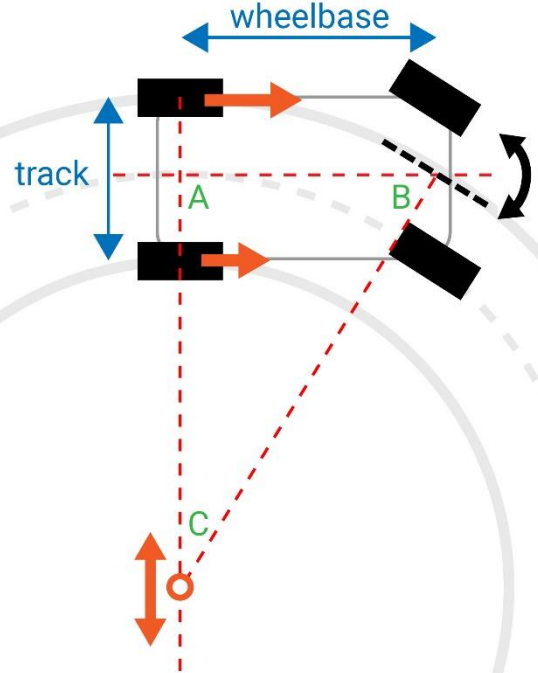
source

distance AC:
use triangulation --> tan = o/a
atan = inversed tangent
AC = atan(ABC) * AB;
ABC = 90 - steering_angle;
NOTE: atan function uses radians!
so ABC = 1,571 - steering_angle / 57,296;
AB = wheelbase;
AC = atan(1,571 - steering_angle / 57,296) * wheelbase;

ratio Left wheel:
radius_left / radius_center
radius_left = AC + 0,5 * track;
radius_center = AC;
ratio = (AC + 0,5 * track) / AC;

ratio Right wheel:
radius_right / radius_center
radius_right = AC - 0,5 * track;
ratio = (AC - 0,5 * track) / AC;

NOTE:
depending on which steering direction relates to a positive steering angle, left and right ratios should be swapped.



Used in **`rover_bt_dabble.ino`**

Differential Drive

Credits: Thimo Willems

CALCULATE MOTOR DEGREES

TO DRIVE A GIVEN DISTANCE

Let's drive 1m

$$C_w = D_w \times \pi \qquad degrees = \frac{distance}{C_w} \times 360$$

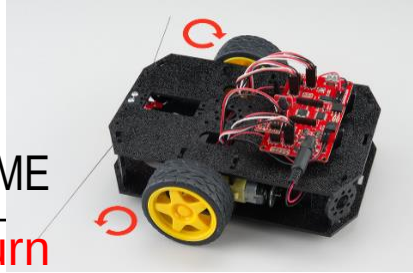
```
#define WHEEL_DIAM 4.96 // wheel diameter in cm

double circumference = WHEEL_DIAM * PI; // value of PI is build-in definition

unsigned int distance = 100; // distance to travel in cm

unsigned long degrees = (distance / circumference) * 360;

evshield.bank_a.motorRunDegrees(SH_Motor_Both, SH_Direction_Reverse, SH_Speed_Medium,
          degrees, SH_Completion_Wait_For, SH_Next_Action_Brake );
```

Run the motor for the calculated amount of degrees

Download example **evshield_drive_1m.ino**

MAKE 90-DEGREE TURN

SPIN WHEELS IN OPPOSITE DIRECTION AT SAME TIME



= point turn

Important: commands should run at same time! How? For first command use **SH_Completion_Dont_Wait**

```
Serial.println("go left");
evshield.bank_a.motorRunDegrees(SH_Motor_2, SH_Direction_Forward, SH_Speed_Medium,
254, SH_Completion_Dont_Wait, SH_Next_Action_Float);

evshield.bank_a.motorRunDegrees(SH_Motor_1, SH_Direction_Reverse, SH_Speed_Medium,
254, SH_Completion_Wait_For, SH_Next_Action_Float);
```

Degrees (in this example: 254) depends on wheel diameter and trackwidth of your robot! See next slide

```
Serial.println("go right");
evshield.bank_a.motorRunDegrees(SH_Motor_1, SH_Direction_Forward, SH_Speed_Medium, 254,
SH_Completion_Dont_Wait, SH_Next_Action_Float);

evshield.bank_a.motorRunDegrees(SH_Motor_2, SH_Direction_Reverse, SH_Speed_Medium, 254,
SH_Completion_Wait_For, SH_Next_Action_Float);
```

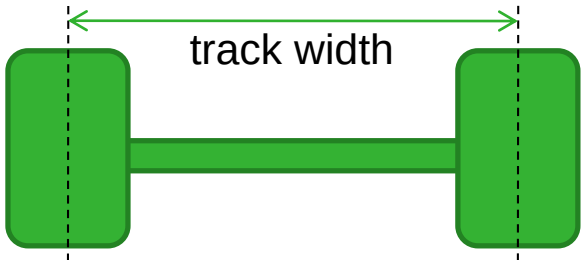
swing turn: rotate one wheel and stop (or slow) the other

CALCULATE DEGREES POINT TURN

$$Rot = n \times \frac{W_t}{D_w}$$

Where

Rot = rotate motor in degrees
n = degree of turn
W_t = track width
D_w = diameter of wheel



Make a 90 degrees turn: wheel diameter=4.96, trackwidth=14:

$$Rot = 90 \times \frac{14}{4.96} = 254$$

Tip: try to make a function **`turn()`**, similar to the function **`driveDistance()`** in example **`evshield_drive_1m.ino`**

Source for calculation

EVSHIELD TIPS & TRICKS

- With 3 or 4 motors connected, the EVShield might sometimes behave strange/appears buggy
- Try:
  - Disconnect all power
  - Test on batteries only (no USB cable connected)
  - (temporary) remove all sensors
  - Switch ports eg. connect motor on M1 > to M2

COLOR SENSORS

Read this info on how to use it

```
color = sensor.readColor();
```

Method **readColor()** returns color-index, eg. 1 = black, 2 = red …

NXT Color Sensor

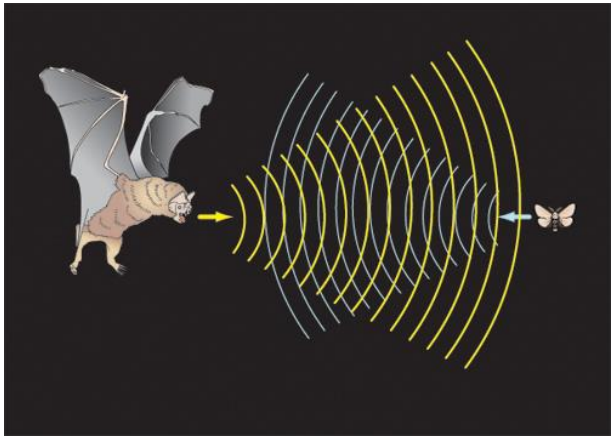
HiTechnic Color Sensor

Read this info on how to use it

EV3 Light/Color Sensor

ULTRASONIC SENSOR

- Uses ultrasonic sound waves to determine range of object (echo-location)
- Range 5-250cm… or more
- Send a ‘ping’… wait for return, measure time to get distance



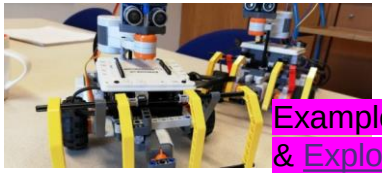
NXT-version cannot be used with EVShield

Basic Arduino module

EV3 version

ULTRASONIC SENSOR

CODE EXAMPLE



Example: Rover & Explorer robot

Basic Arduino module

Uses 'NewPing' library

```
#include <NewPing.h>

// define to which pins the sensor is connected:
#define TRIGGER_PIN  3
#define ECHO_PIN     5
#define MAX_DISTANCE 300 // Maximum distance we want to ping for (in centimeters).
                         //Maximum sensor distance is rated at 400-500cm.

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // NewPing setup of pins and maximum distance.

unsigned int distance = sonar.ping_cm(); // read distance from ultrasonic sensor
if ( distance > 0 && distance < 30 ) {   // is it larger than 0 and smaller than 30 (cm)?
  Serial.println("object detected");
}
```

A valid distance must be > 0

EV3 version

See EVShield examples

ROTATION SENSOR

- Motor has build-in rotation sensor

```
degrees = evshield.bank_a.motorGetEncoderPosition(SH_Motor_1);
```

Download example **evshield_motorGetEncoderPosition.ino**

GYRO SENSOR

- Combination of detection of angle and acceleration
- Can be used to create balancing robots

See EVShield examples

Gyro Boy, can be build with EV3 kit, which can be borrowed

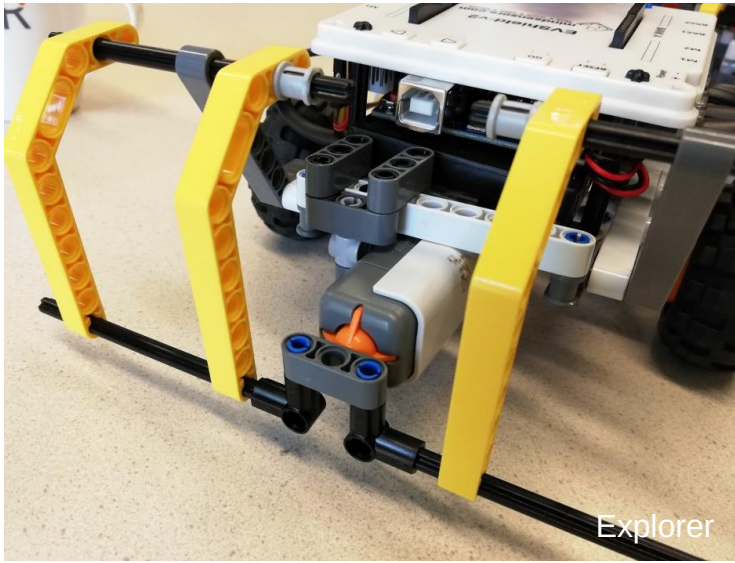
youtube.com/watch


Oops...
source

BUMPER

File > Examples > EVShield > EVShield_tests > nxt_touch

- **`loop()`**: check multiple ‘events’
- timing of **`loop()`**
- Look at example given for Explorer or Rover (`rover_bt_dabble.ino`)




```
if (drive && forward) { // if we are driving forward...
  if (myTouch.isPressed()) { // if we bump into something
     // stop and back-up
     Serial.println("bump");
     reverseTurn();
  }
}
```

BUMPED… BACKUP AND TURN

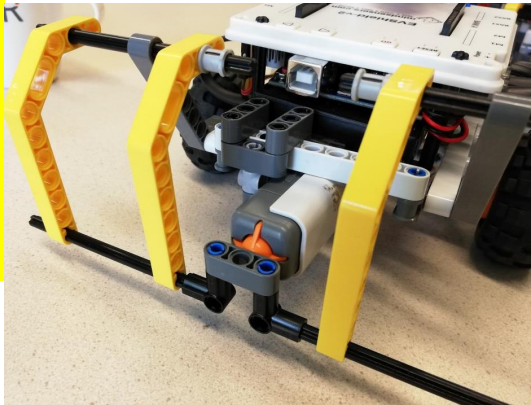
```
void reverseTurn() {
}
```

Can you create a function that will make the robot go backwards and turn?

- Pseudo code:

```
void reverseTurn() {
  stop (both motors)
  drive backward a little
  make a 90 degree turn
  continue driving
}
```

```
void reverseTurn() {
  // stop both motors
  evshield.bank_a.motorStop(SH_Motor_Both, SH_Next_Action_Brake);

  // drive backward a little
  evshield.bank_a.motorRunUnlimited( SH_Motor_Both, SH_Direction_Forward, SH_Speed_Medium );

  // make a 90 degree turn
  turn(90);

  // continue driving
  evshield.bank_a.motorRunUnlimited( SH_Motor_Both, SH_Direction_Reverse, SH_Speed_Medium );
}
```



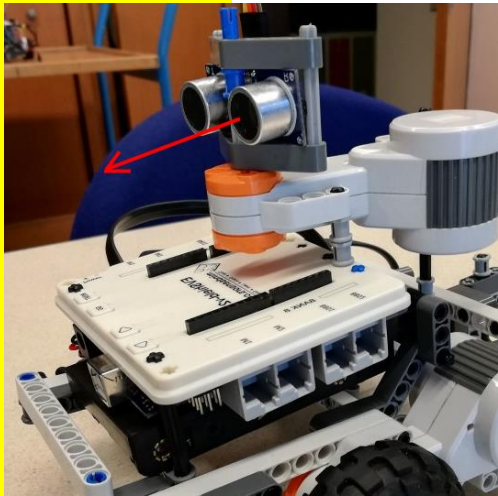
SEE AN OBSTACLE… FIND A WAY OUT?

```
void findWay() {
}
```

Can you create a function that will find a way out?

```
void findWay() {
  // stop (both motors)
  evshield.bank_a.motorStop(SH_Motor_Both, SH_Next_Action_Brake);

  // look left:
  // Use evshield.bank_b.motorRunDegrees(…) to turn the motor with the ultrasonic sensor 90 degrees
  // take a reading of the ultrasonic sensor
  unsigned int left = sonar.ping_cm(); // read distance from ultrasonic sensor

  // look to the right (turn 180 degrees in other direction)
  // take another reading
  unsigned int right = sonar.ping_cm(); // read distance from ultrasonic sensor

  // rotate the motor back in original position
  // decide where to go:
  if (left>right) { // go left
      // turn 90 degrees to left
  }
  else { // go right
      // turn 90 degrees to right
  }
  // continue driving
}
```

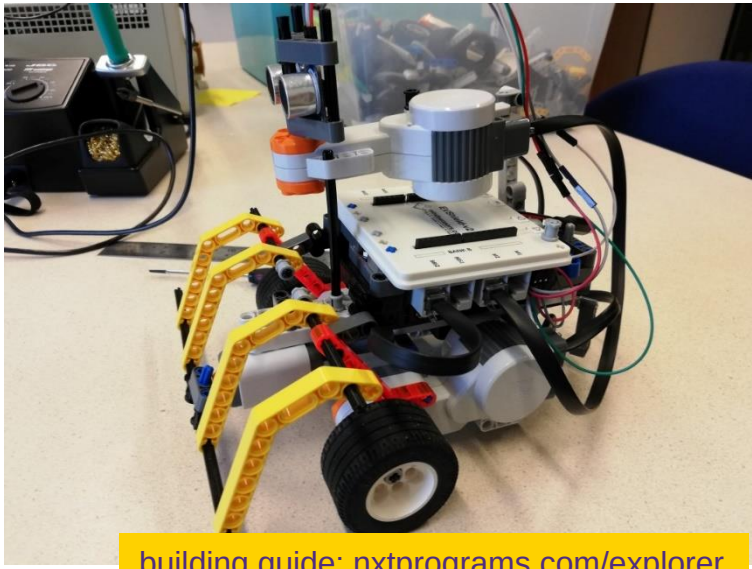
Comments… are ‘left-overs’ from pseudo code



ASSIGNMENT: PRACTICE MORE LEGO BUILDING

OPTION 1: BUILD EXPLORER

- Build Explorer robot from scratch, eg. from this guide: **nxtprograms.com/explorer**
- Be creative: make some adjustments to fit the EVShield pack



building guide: nxtprograms.com/explorer

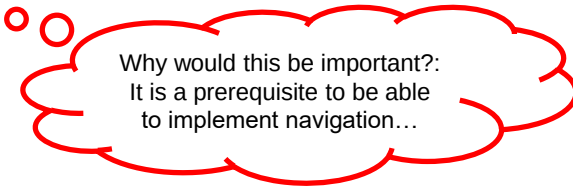
Warning: you must keep a Lego car (Rover or Explorer), as we will use that for Assignment 4 of Application Development

If your kit is missing pieces/materials, you can get replacements from teacher!

ASSIGNMENT: IMPROVE ROVER CAR

OPTION 2: IMPROVE CAR

- Read **step 7** “Improve the car” of the Rover Car tutorial. Execute steps outlined there (there are links to separate tutorials!).
- Make the car (Rover or Explorer) capable of avoiding obstacles (see slides 22-25)
- Make it able to drive measured distances (eg. drive exactly 2m) and make precise turns (eg. make a 90 degree turn around its own axis) (slides 14-15)



Warning: please keep a Lego car (Rover or Explorer), as we will use that for Assignment 4 of Application Development

ASSIGNMENT: BUILD A COLOR XYLOPHONE

OPTION 3: WORK WITH ARDUINO ONLY

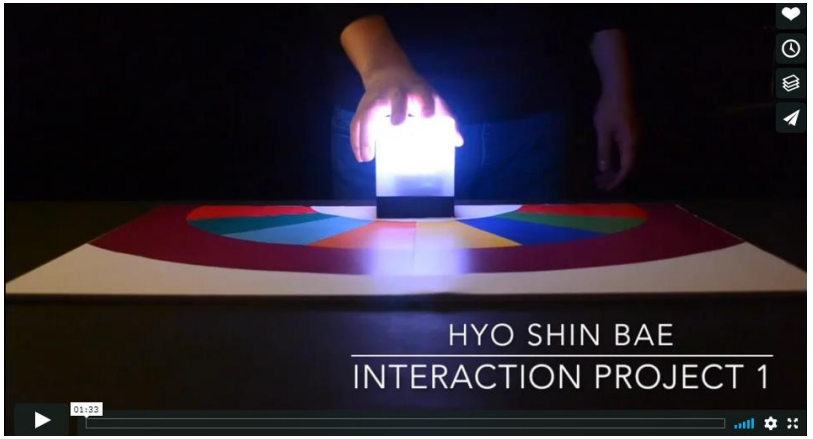
1. Follow step 1 of tutorial "Using the TCS3200 color sensor"
2. Follow tutorial "Sounds and music with Arduino" - choose a method to make sound
3. Combine the two to make a Color Xylophone. Inspiration:

For 1 and 2: use an Arduino Nano if doing this separate from the Rover car

Tip: split group in half: one team does step 1, other 2

slides @ vanslooten.com/appdev

"Xylophone Light - Sound of Color"

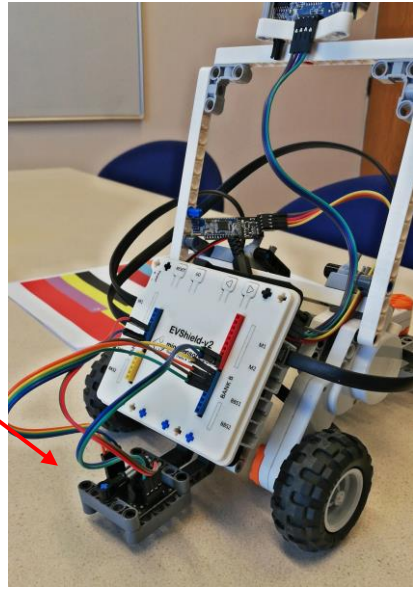


ASSIGNMENT: CHALLENGES

GOAL: DIVIDE TASKS SMART, LEARN AS MUCH AS POSSIBLE

Tip

- As a group, you can do multiple options: e.g. half of your group does option 2, other half option 3
- You can do more: combine option 2 & 3: make a car which drives over colored tiles on the floor to make music! (color sensor tutorial has a second step which shows how to mount a color sensor to the car)



This assignment is a mandatory part of Application Development.
As a group, do at least one option and demonstrate it to receive a ‘pass’.

If your kit is missing pieces/materials, you can get replacements from teacher!