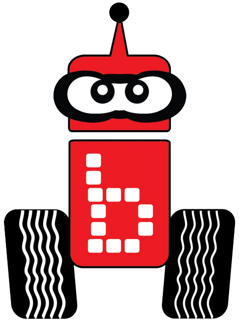


Motor Position Counters

- Key Concepts
  - Understand what motor position counters are and how to use them.
- Pacing
  - Over several class periods.

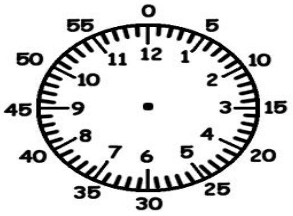
KISS
INSTITUTE FOR
PRACTICAL
ROBOTICS

Understanding Motor Position Counter Activity 1

Read and discuss this slide.

Each motor used by your robot has a built-in motor position counter, which you can use to calculate the distance traveled by the robot!

Similar to how a clock is divided into 60-second intervals (ticks).



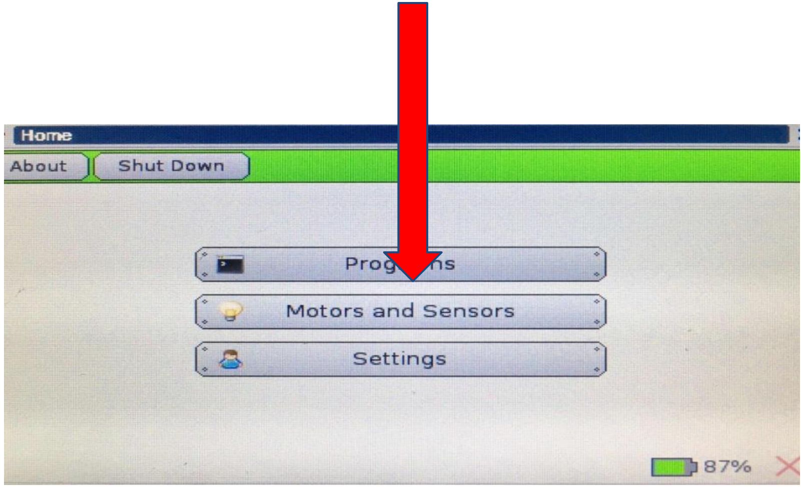
- Motor position is measured in "**ticks**".

- Your motors have ***approximately* 1400 ticks per *revolution*** (This is not related to the size of your wheel).

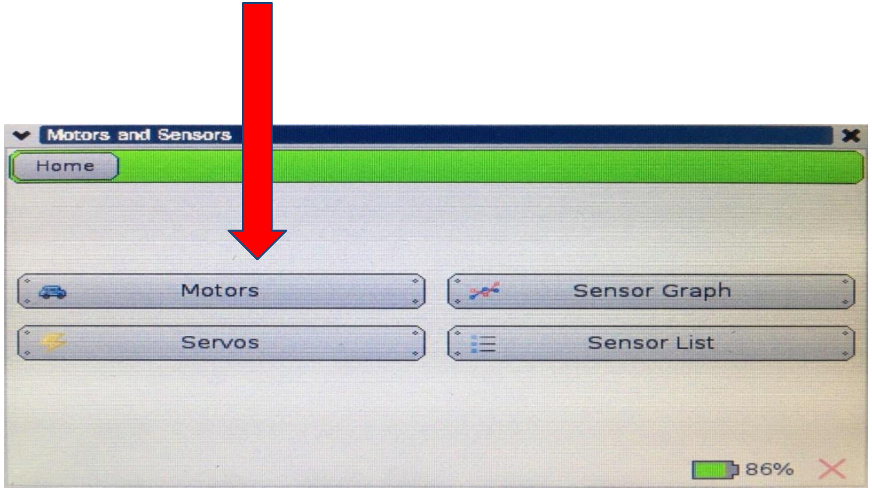
KISS INSTITUTE FOR PRACTICAL ROBOTICS

Using the Robot Motor Screen to See Motor Positions

Select Motors and Sensors



Select Motors



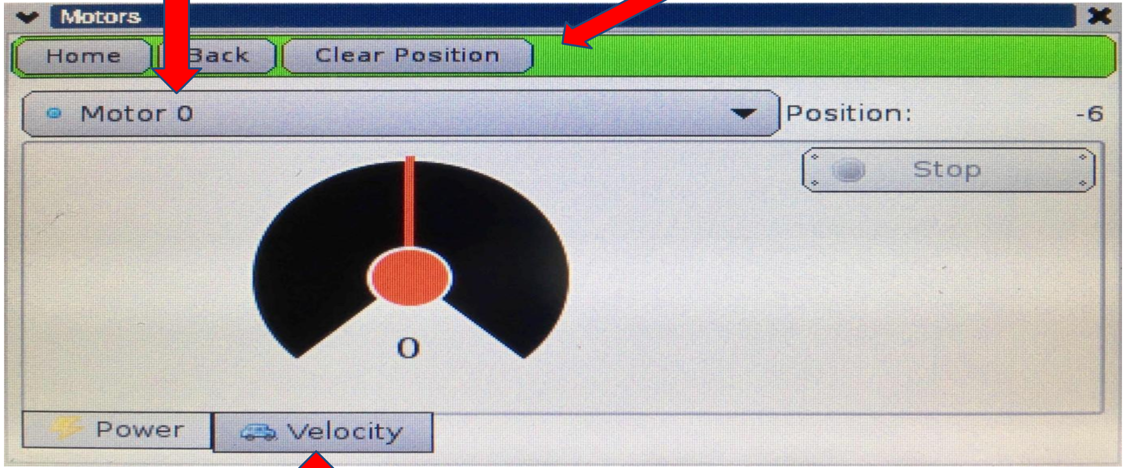
KISS INSTITUTE FOR PRACTICAL ROBOTICS

Using the Robot Motor Screen to See Motor Positions Activity 2

Select Motor Port (allows you to select the motor of your choice)

Clears position of motor

Motor Position in "Ticks"



Motor 0

Position: -6

Clear the motor position and then using your hand to rotate the robot's wheel, plugged into port 0, and watch the position counter. What happens if you turn the wheel in the opposite direction

Make sure Velocity is selected

You can also place your robot on a surface and roll it forward to measure the # ticks from a starting position to another location or object

Activity 1

1. Open widget on the robot.
2. Place robot on the start line.
3. Slowly push the robot forward to circle 9.
4. Record the ticks on your motor.

Motor Position Counter Functions

Read and discuss new functions you use:

Motor Port #

(#0 – 3)

```
get_motor_position_counter(3);  // or gmpc ();
```

- Tells the robot the number of ticks the motor on port #3 has rotated.

Motor Port #

(#0 – 3)

```
clear_motor_position_counter(3);  // or cmpc ();
```

- Resets the robot’s tick counter to 0 for the motor on port #3.

Proceed to the next slide…...

Activity 2

1. Create a program that:
   a. Moves forward to your recorded ticks and then stops (before circle 9).
2. Psuedocode:
   a. Clear motor position
   b. Create a while loop that will get motor position and drive forward until it reaches your record tick number
   c. Exit loop

Motor Position Counter

```
#include <kipr/botball.h>

int main()
{
  clear_motor_position_counter(3);

  while (get_motor_position_counter(3) < 2700)
  {
     motor (3,100);
     motor (0,100);
  }

  ao();

  return 0;
}
```

Motor Position Problem Solving

Did your robot stop at the number of ticks you put into your code? It most likely went further than you wanted it to go.

●

•Hint- Go to the motor widget and see how many ticks the motor actually turned

•

•The reason the robot traveled further is that the robot has inertia (It is moving and even though the motor position loop shuts the motors down the robot will still coast until it actually stops)

•

•If your target distance was 11500 ticks and when you run the program it acutualy goes to 12110 ticks then the overage is 12110 – 11500 = 610

•

•Change your program so that the target distance is 11500- 610 = 10890

•You can also do the math right in the code

```
while (get_motor_position_counter(3) < 11500 - (12110 - 11500))
```

Activity 3

1. Place a can on circle 9.
2. Use your new functions to go out and grab the can.
3. Go to the next slide.

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Motor Position Counter Pseudocode Sample

1. Find the ticks on one of your motors to reach circle 9.
2. Go to the widget and select one motor.
3. Roll it until you reach the outside of circle 9.
4. Record the ticks.
5. Write a program that:
    a. Clears the position of one motor.
    b. Gets position counter less than your recorded ticks (<2700).
    c. Ask it to move forward to a certain position.
    d. Then exit the loop and pick up the can.
    e. Think about where you need to place set your servos.
    f. Return to start finish.

KISS INSTITUTE FOR PRACTICAL ROBOTICS

```
#include <kipr/botball.h>
arm = 3
//claw = 0
//open = 1500
horizontal = 145

int main()
{
enable_servos();
set_servo_position (arm,horizontal);//horizontal
msleep (500);
set_servo_position (0,1500);//open
msleep (500);
  clear_motor_position_counter(3);
  while (get_motor_position_counter(3) < 2700)
  {
     motor(3, 100);
       motor (0,100);
  }
ao:
set_servo_position (0,100);
msleep (500);
while (get_motor_position_counter(3) > 0)
  {
      motor(3, -100);
       motor (0,-100);
  }

  ao();
  return 0;
}
```

Motor Position Counters
Activity 4

Goal: Write a program that drives your robot forward for 4 *motor revolutions*, and then stops.

- Remember: there are 1400 motor ticks per revolution.

1. Use code planning paper to plan the steps in your code to move your robot forward approximately 4 revolution.
2. Write the code and run the program.

Hint: Your code will need a loop