P.O.T.A.T.O.E.S.
PEOPLE OF TOMORROW APPLYING THEIR OUTSTANDING ENGINEERING SKILLS

VEX Robotics Competition (VRC)

Phase 5

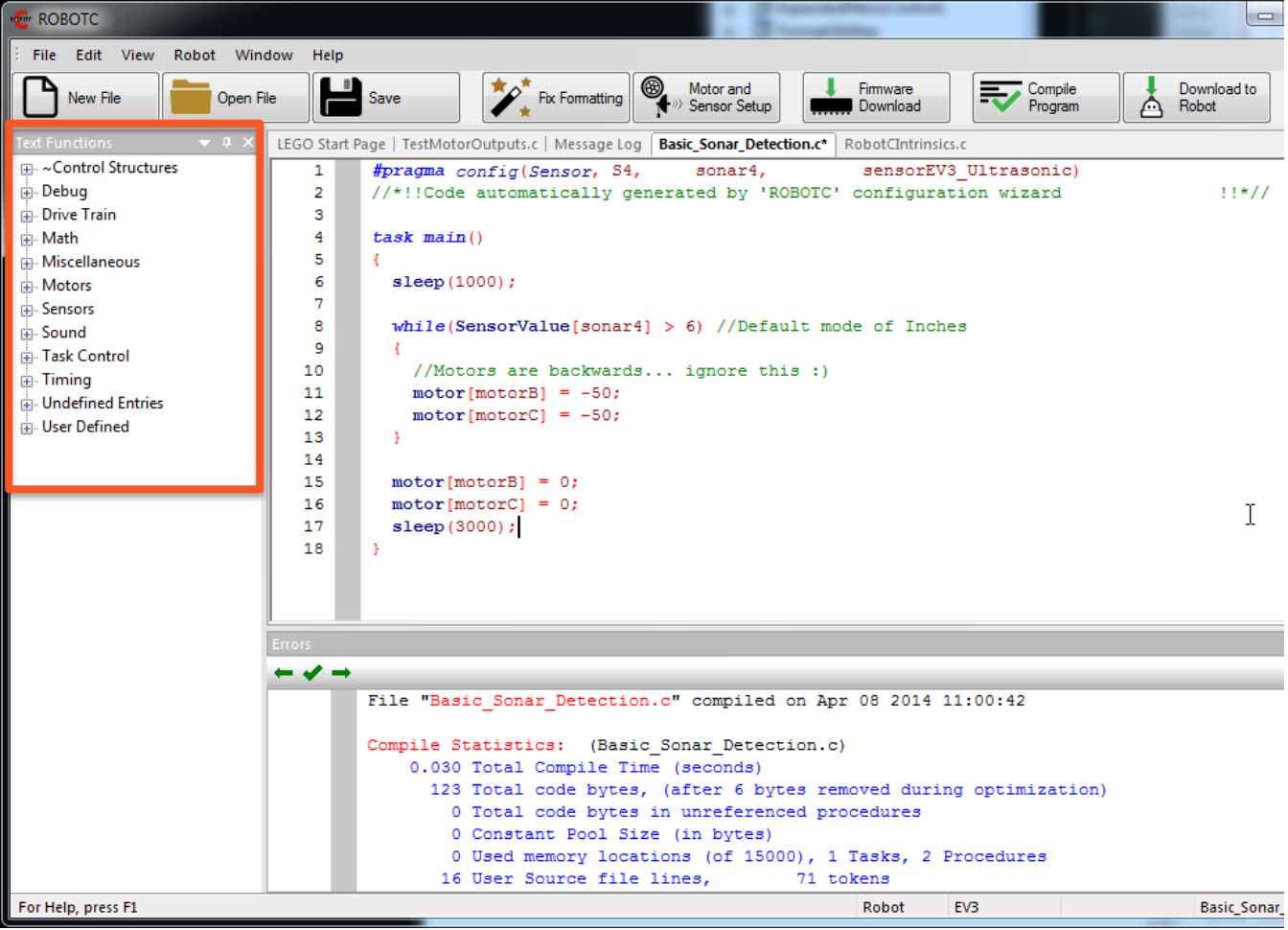
Edwin Kofler and Eli Helmer

Agenda

- Programming
- Sensors
- Programming test robots
- Programming your Robot

Programming Language: RobotC

- C-based programming language
- Real-time debugger
- Only available at the Engineering Lab computers



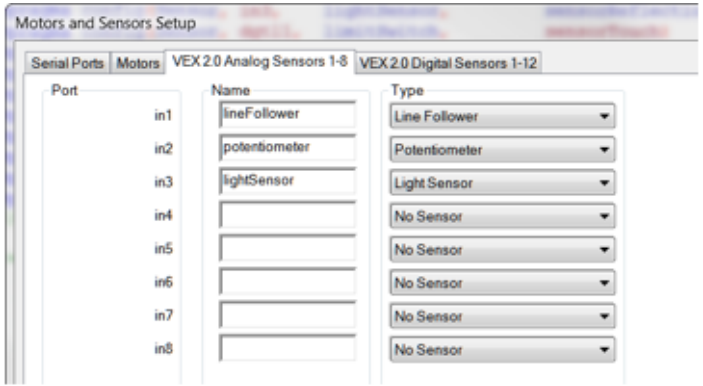
- Text Functions => View all programming commands

P.O.T.A.T.O.E.S.
PEOPLE OF TOMORROW APPLYING THEIR OUTSTANDING ENGINEERING SKILLS

Using Menus

- Firmware Download
  => Check for updates
- Compile Program
  => Compiles the program in a readable format for the Cortex and checks for errors
- Download to Robot
  => Downloads code to robot

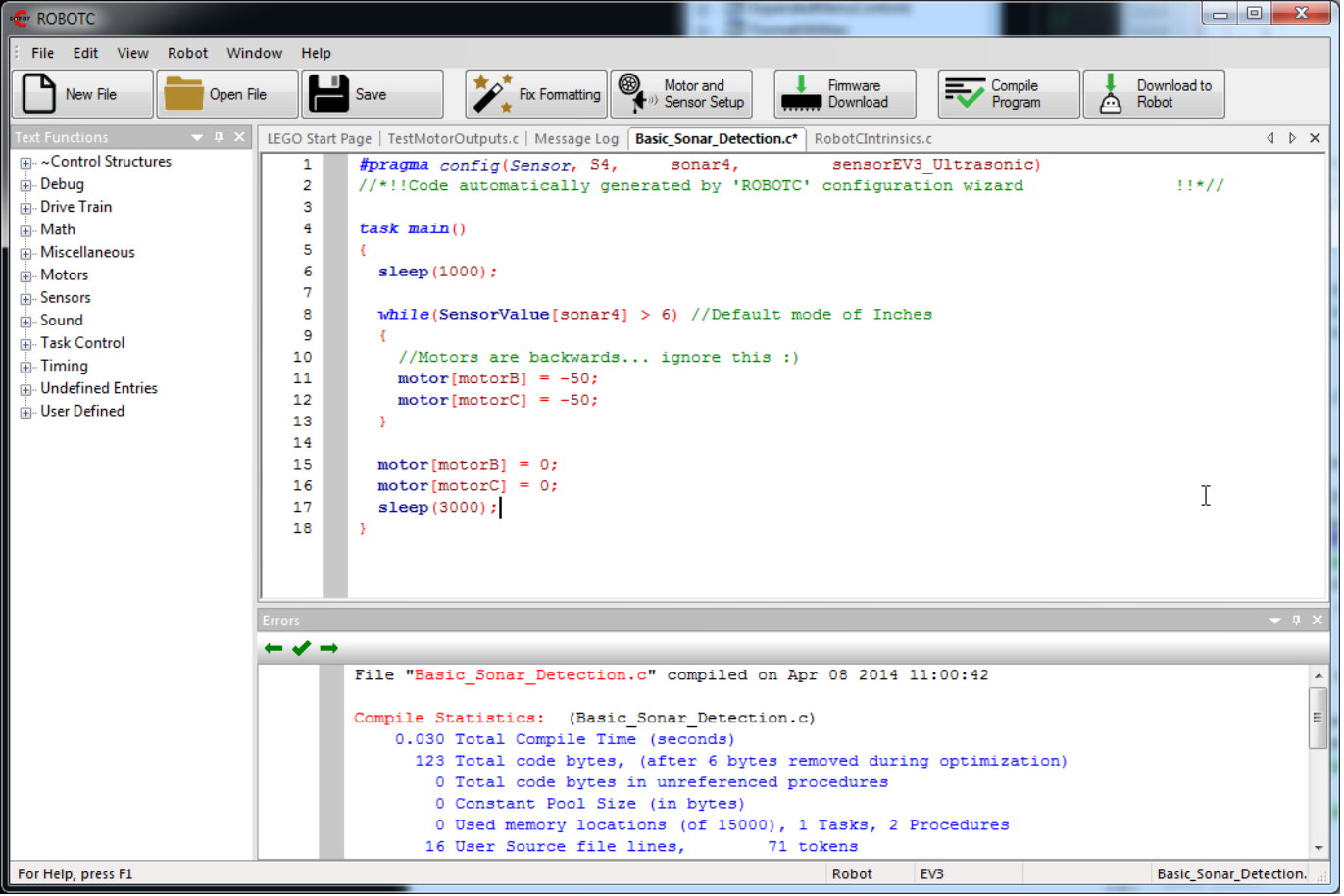
Configuring Ports with Motors



- If you want send information to or from a device you first must assign that device to a certain port. Then, you must give that device a name
- Motor and Sensor Setup => Edit the names and locations of the VEX devices' (analog, digital, or motor) ports

P.O.T.A.T.O.E.S.
PEOPLE OF TOMORROW APPLYING THEIR OUTSTANDING ENGINEERING SKILLS

Debugger Menu



- Once you click => Download to Robot, debugger menu shows up
- Click on sensors menu, and it will show the input received from sensors and the port the sensor is connected to

- For more info on RobotC: http://cdn.robotc.net/pdfs/natural-language/Natural_Language_Cortex.pdf

```
startMotor(port8, 63);
untilPotentiometerGreaterThan(4000, in4);
stop();
```

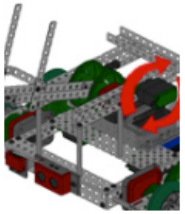
Steps to Program A VEX Robot

- Connect the motors and sensors in various ports of the robot (also configure the ports)
- Write program as text, Referencing the different ports
- Compile program and check/fix programming errors
- Download program to robot
- If program works as planned, make the robot work with a VEX controller

VEX Sensors and their Command Implementation

Motors and Servos

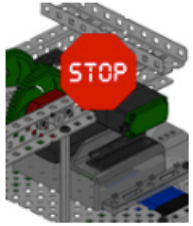
Start Motor
Set a motor to a speed.



Command:

```
startMotor(motor, speed);
```

Stop Motor
Stops a motor.



Command:

```
stopMotor(motor);
```

Set Servo
Set a servo to a desired position.

Command:

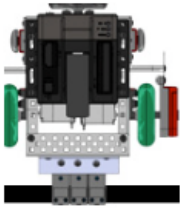
```
setServo(servo, position);
```

Line Tracker

- Remember the semi- colons

Until Dark

The robot continues what it was doing until the line tracking sensor reads a value darker than a specified threshold.

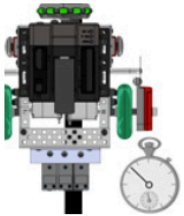


Command:

```
untilDark(threshold, sensorPort);
```

Line Track for Time

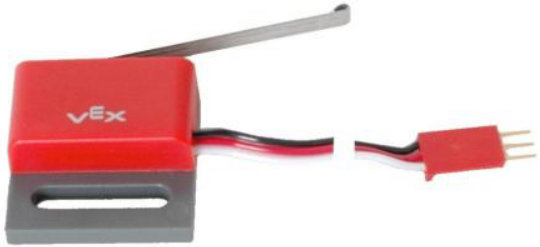
The robot will track a dark line on a light surface for a specified time in seconds.



Command:

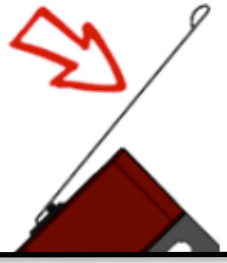
```
lineTrackForTime(time, threshold, sensorLeft, sensorCenter, sensorRight);
```

Limit Switch and Bumper Switch



Until Touch

The robot continues what it was doing until the touch sensor is pressed in.



Command:

```
untilTouch(sensorPort);
```

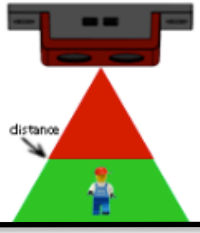
Until Bump

The robot continues what it was doing until the touch sensor is pressed in and then released out.
(A delay time in milliseconds can be specified as well.)

Command:

```
untilBump(sensorPort, delayTimeMS);
```

Sonar and Ultrasonic Encoder



Until Sonar Greater Than

The robot continues what it was doing until the sonar sensor reads a value greater than a set distance in centimeters.

Command:

```
untilSonarGreaterThan(distance, sensorPort);
```

Until Rotations

The robot continues what it was doing until the quadrature encoder rotations reach the desired value.

Command:

```
untilRotations(rotations, sensorPort);
```

Potentiometer, Flashlight, and LED

Until Potentiometer Greater Than
The robot continues what it was doing until the potentiometer sensor reads a value greater than a set position.

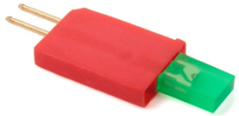
Command:

```
untilPotentiometerGreaterThan(position, sensorPort);
```

Flashlight ON
Turn a VEX Flashlight in a specified motor-port ON at a specified brightness.

Command:

```
turnFlashlightOn(motorPort, brightness);
```



LED ON
Turn an LED in a specified digital-port ON.

Command:

```
turnLEDOn(sensorPort);
```

Agenda

- Programming test robots
  - Program robot to move straight, stop, turn left, then move straight again
  - Program robot to move in a circle
  - Program robot to follow a line
- Programming your Robot

End of Phase 5