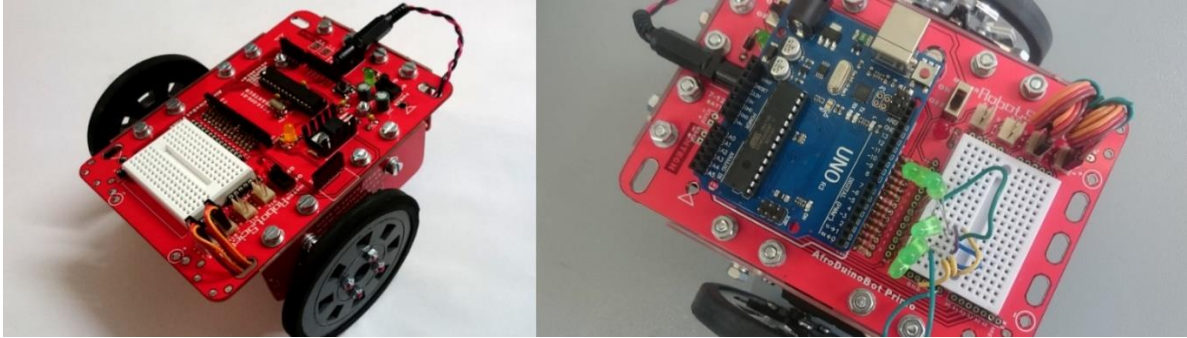


ROBOTICS 101



HANDBOOK 1:

BASIC ELECTRONICS PROGRAMMING MICROCONTROLLERS ROBOTICS and MECATRONICS

COACH MICHAEL ETTERS HANK

An introduction to desktop robotics and mecatronics, including how to source the parts required to assemble your first robot, how to solder the electronics and build the chassis, how to write and upload text-based program code, how to wire up and understand sensors, how to build and prepare for robotics competitions and a career in engineering

Title: Robotics 101: Handbook 1

Author: Michael Ettershank

Publisher: Michael Ettershank

Distributor: Mantech.co.za

+27 82 962 2772

info@robotscience.co.za

COPYRIGHT NOTICE

Please note that the entire contents of this training manual remain the intellectual property of the author, and may not be copied by physical or electronic means, or any part thereof stored in an electronic retrieval system or public or private local area network without prior written permission.

Failure to observe the above warning will result in civil and criminal legal proceedings.

WHISTLEBLOWER POLICY

The age of the internet has seen wholesale copying of entire textbooks, making it very difficult to justify the cost of creating quality training. Copying intellectual property [IP] and copyrighted texts is not a 'victimless crime' as it directly hurts those who create quality content, discouraging development of QUALITY training.

Progressive countries have realised the value of encouraging folks to create quality training, and will award damages and legal costs when the law is broken. If you become aware this work has been copied illegally, please delete it and obtain a legal copy so we can produce more quality training. If you become aware of an organisation using this textbook illegally and you notify us, you will be awarded 50% of any damages that we receive from the ensuing lawsuit to prevent illegal copying.

If there are topics you would like us to include in future volumes in this training series, or if you are aware of an organisation using this textbook illegally, please contact us on info@robotscience.co.za.

ROBOTICS 101 HANDBOOK 1



TABLE OF CONTENTS

WORKSHOP 1: Getting Started

WORKSHOP 2: Prototyping Breadboards

WORKSHOP 3: Soldering 1

WORKSHOP 4: Soldering 2

WORKSHOP 5: Sensor Basics 1

WORKSHOP 6: Sensor Basics 2

WORKSHOP 7 and 8: Electronics Build

WORKSHOP 9: Robot Assembly

WORKSHOP 10: Dead Reckoning

WORKSHOP 11: Maze Competition

WORKSHOP 12: Fault Finding

WORKSHOP 1

Introduction to Robotics

WORKSHOP 1.1: Getting Started

WORKSHOP 1.2: Hello Text Software

WORKSHOP 1.3: Flash LED on D13

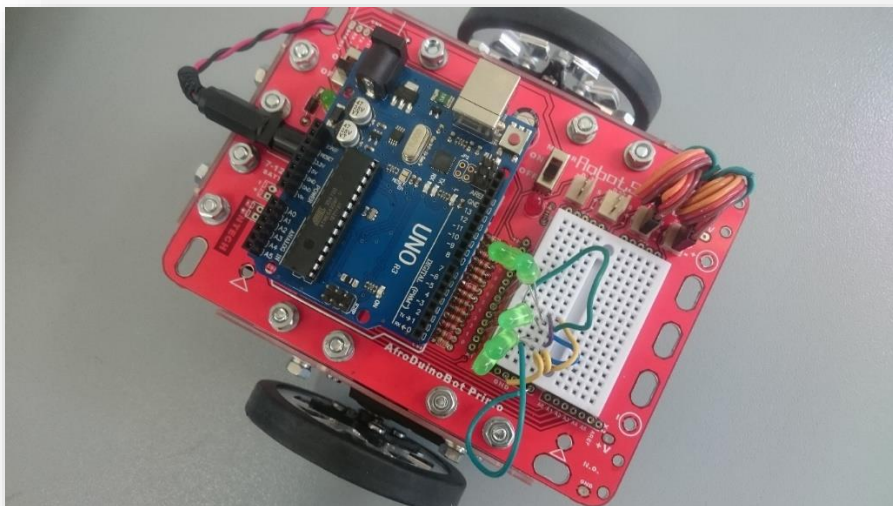
WORKSHOP 1.4: Counting

WORKSHOP 1.5: Robot Motors FWD

WORKSHOP 1.6: Motors LEFT and RIGHT

WORKSHOP 1.7: Calibrate Motors

WORKSHOP 1.8: Motor Calibration in Software



WORKSHOP 2

BREADBOARD CIRCUITS

WORKSHOP 2.1: Mini Breadboard

WORKSHOP 2.2: Bigger Breadboards

WORKSHOP 2.3: Hand Tools

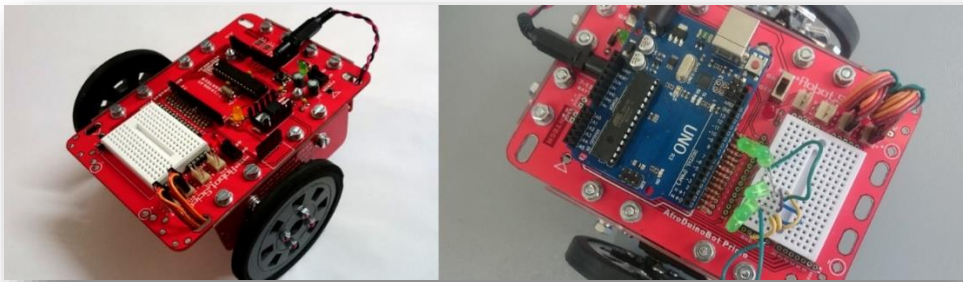
WORKSHOP 2.4: Circuit on Breadboard: Flasher

WORKSHOP 2.5: Breadboard Switch Circuit

WORKSHOP 2.6: Breadboard LED Circuit

WORKSHOP 2.7: Sensor on Breadboard

WORKSHOP 2.8: Two Sensors on Breadboard



WORKSHOP 3

SOLDERING FOR ELECTRONICS: 1

WORKSHOP 3.1: Introduction to Soldering

WORKSHOP 3.2: Good Soldering Iron

WORKSHOP 3.3: A Bad Solder Iron: Do not Buy!

WORKSHOP 3.4: Solder Practice: Shapes

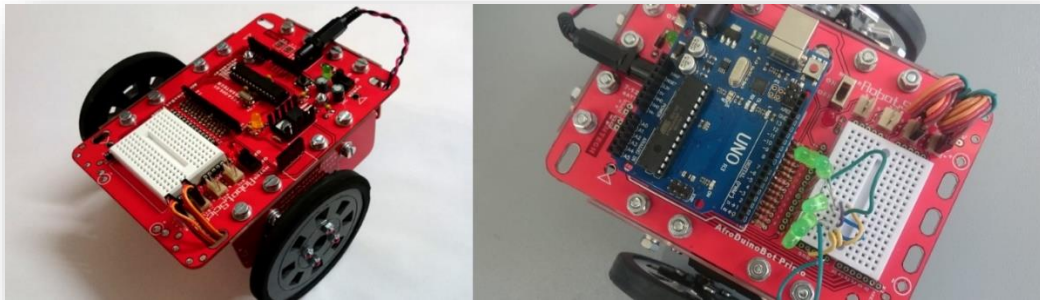
WORKSHOP 3.5: Soldering: Two LED Flasher



WORKSHOP 4

SOLDERING FOR ELECTRONICS: 2

A Two Transistor Flasher on Veroboard



WORKSHOP 5

SENSORS 1: THE BASICS

WORKSHOP 5.1: Binary Electronics

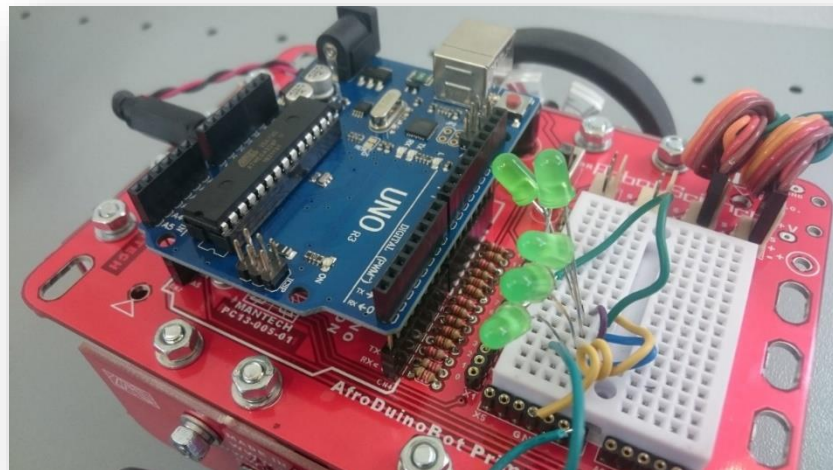
WORKSHOP 5.2: Series and Parallel Circuits

WORKSHOP 5.3: Voltage Divider Circuits

WORKSHOP 5.4: Digital Input: Pull Down Resistor

WORKSHOP 5.5: Input Serial Monitor

WORKSHOP 5.6: Transistor Switch



WORKSHOP 6

SENSORS 2: A LINE FOLLOWING MODULE

WORKSHOP 6.1: Analogue Inputs

WORKSHOP 6.2: The Photo Transistor

WORKSHOP 6.3: Light Sensitive Robot

WORKSHOP 6.4: Line Module

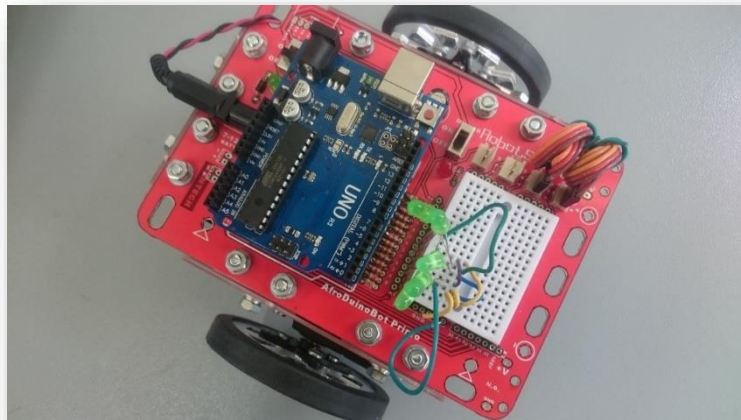
WORKSHOP 6.5: Module Serial Test

WORKSHOP 6.6: Line Follower Explained

WORKSHOP 6.7: Serial Monitor View

WORKSHOP 6.8: Three Sensor Code

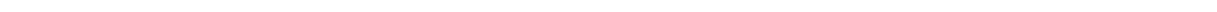
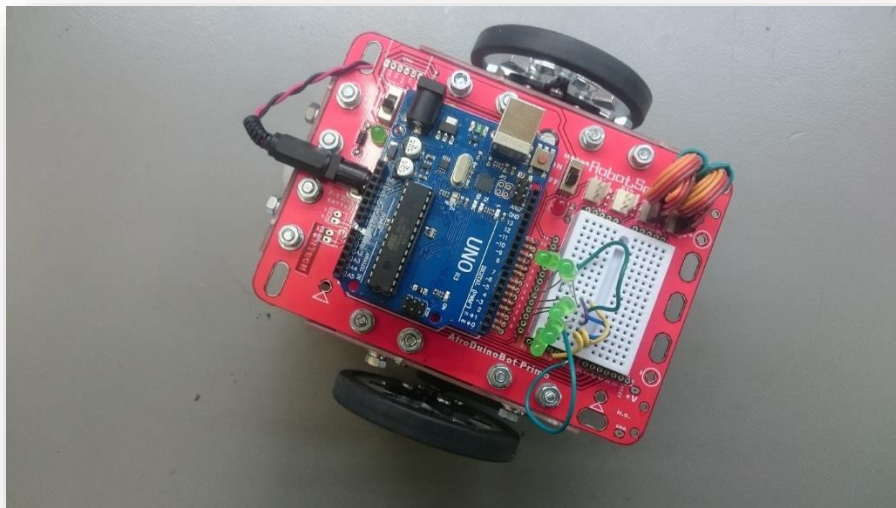
WORKSHOP 6.9: Sensor Indicator LEDs



WORKSHOP 7 + 8

THE CONTROLLER ELECTRONICS

Build the PCB Controller Chassis [Two Sessions]



WORKSHOP 9

ASSEMBLING THE ROBOT

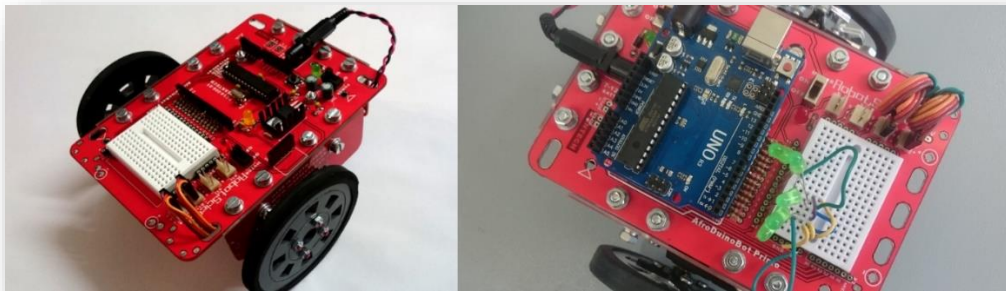
WORKSHOP 9.1: Assemble the Chassis

WORKSHOP 9.2: Assemble the Wheels

WORKSHOP 9.3: Connect the Motors

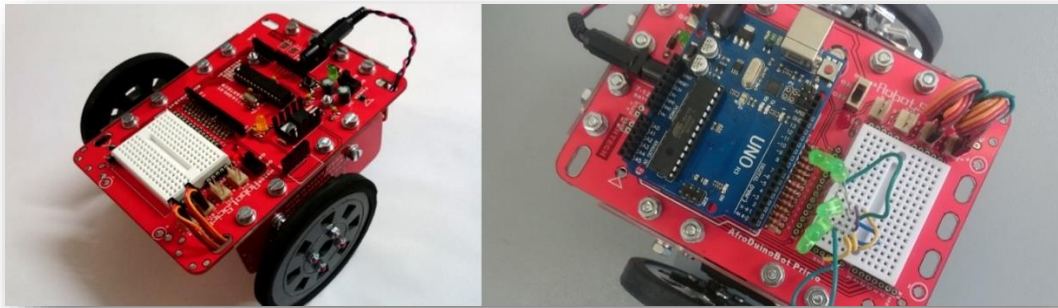
WORKSHOP 9.4: Calibrate the Motors

WORKSHOP 9.5: Test the Motors



WORKSHOP 10

NAVIGATION: DEAD RECKONING



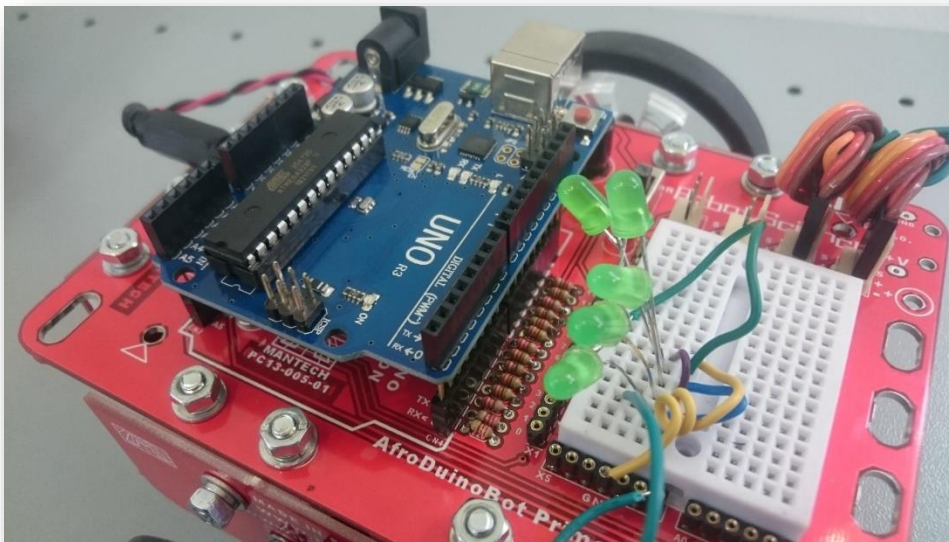
WORKSHOP 11

A DIY ROBOTICS COMPETITION

WORKSHOP 11.1: A Maze with 45 Deg Turns

WORKSHOP 11.2: Robotics Competitions

WORKSHOP 11.3: Competition Rules



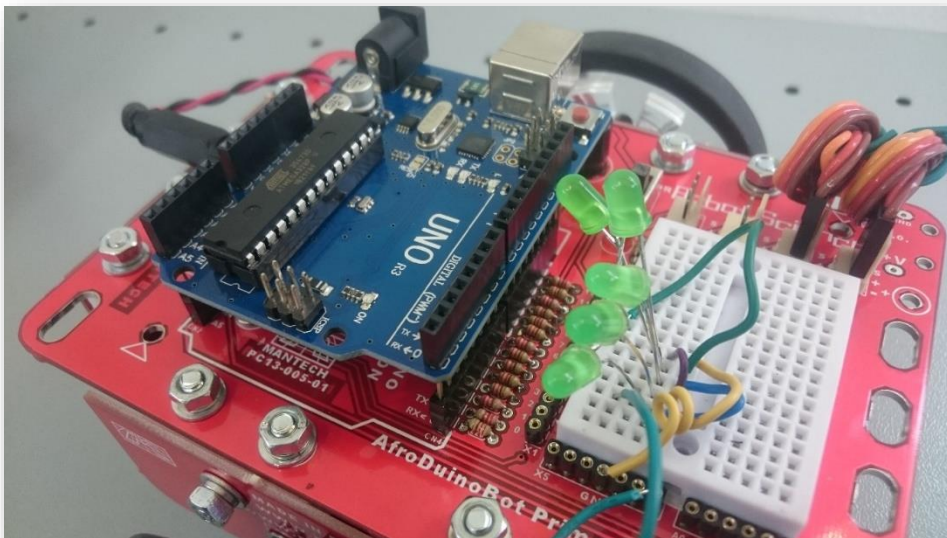
WORKSHOP 12

FAULT FINDING and PARTS LISTs

WORKSHOP 12.1: Fault Finding


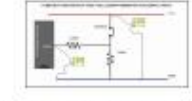
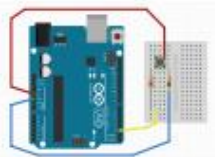
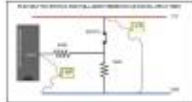


WORKSHOP 12.2: Primo Robot Parts List

WORKSHOP 12.3: Tools List



Thank you for purchasing this robotics textbook. By making this purchase you are supporting the production of more exciting textbooks in this series.

In **ROBOTICS 101 HANDBOOK 1** you will learn the essential basics that you require to embark on an exciting career in electronics, robotics and mechatronics. We have taken great care that there's not too much theory, and lots of practical learning activities including how to build circuits, how to build and program your first robot. In this series we are planning to teach you about basic machine learning, how to design and 3D print add-on equipment and prepare you for an exciting career.

<p>PAGE 61 ROBOTICS 101 HANDBOOK 1 ©2018 MICHAEL PETERSHANS</p> <p>ROBOTICS 101 WORKSHOP 6.4 DIGITAL INPUTS AND THE PULL-DOWN RESISTOR</p> <p>To be able to attach a sensor to your robot you must understand how microcontroller inputs work. The Arduino Uno and Arduino Pro Mini and many other types of microcontroller have digital and analogue inputs.</p>  <p>DIGITAL AND ANALOGUE INPUTS</p> <p>The two types of inputs, which is easiest to understand, is called a DIGITAL IN PUT. To read, we are going to look at how digital inputs work, and we will look at an example of how to do this. The second and more sophisticated type of input is the ANALOGUE INPUT which we will look at a little later.</p> <p>Whatever type of controller board you have, all robots have inputs and outputs. Outside can be sensors or actuators, they require one version of various types. The most basic digital sensor you can use can be a switch or a limit or a simple physical switch. A digital activated push-button switch can be used to turn your robot into a "digital sensor". Another simple sensor is a digital sensor system could be a non-contact switch obtained for a float in a water tank, which activates a pump or when the water level gets too low and switches the pump off again once the water is full again.</p> <p>Whether a digital sensor system is simple, as in a switch, or a sophisticated like for example a magnetically activated proximity switch, it is important to understand that there are very few possibilities being ON or OFF. How does digital input then deal with something like temperature where there is a whole range of values? Quite simply a digital input has no way to detect such something as temperature sensor however a digital input pin can receive a temperature value from a temperature sensor with an ADC which (depending on digital controller) that change a temperature reading into a string of binary code.</p>  <p>What you will notice in the above two diagrams, is that when the switch contacts are closed, in other words the circuit as it is completed, the digital input pin is pulled up to HIGH. When the switch contacts are open, the PULL-DOWN resistor pulls the digital input pin down to 0. The PULL-DOWN resistor value is relatively high, but if a voltage is still lower than the open state of the switch where resistance is relatively high.</p> <p><i>*Reading in the way helps to describe both the word analogue, both terms are the same. However, if you spell the word like your English teacher taught you in your program (the code might not work!)</i></p>	<p>PAGE 62 ROBOTICS 101 HANDBOOK 1 ©2018 MICHAEL PETERSHANS</p> <p>So, the way a digital input pin on a microcontroller works is the code checks whether the input pin is high or low. Below here, in the circuit diagram we looked at on the previous page, there is a LED connected. That way we can see when the microcontroller has received high voltage (light), that might change it.</p>  <p>The above 4 wiring diagrams (a wonderful free software anyone can download and use) show how to connect everything up to test a digital input.</p> <p>Let's take a look at the code that would be required to check what the "level" (high or low) of a digital input pin of the microcontroller is currently at and when the LED connected to D13 on the UNO.</p> <pre data-bbox="853 974 1212 1131">int pinNumber = 0; int ledPin(); pinMode(0, OUTPUT); pinMode(13, OUTPUT); void setup() { pinMode(pinNumber, INPUT); digitalWrite(ledPin, LOW); } void loop() { digitalWrite(ledPin, digitalRead(pinNumber)); delay(500); } // digitalWrite(ledPin, HIGH); // digitalWrite(ledPin, LOW);</pre> <p>The above code will monitor the input state of a switch connected to digital pin D0 and switch an LED that is connected to D13 on digital pin D13. Incidentally, there's an LED connected to D13 on all Arduino development boards including the Arduino Uno and the Pro Mini etc.</p>
<p>PAGE 63 ROBOTICS 101 HANDBOOK 1 ©2018 MICHAEL PETERSHANS</p> <p>ROBOTICS 101 WORKSHOP 6.6 SERIAL MONITOR VIEW A DIGITAL INPUT PINS STATE</p> <p>When you attach a sensor to your robot, it is useful to be able to know what the state of the input pin is at all times whether the system is working properly or not. In the first diagram, below the PULL-DOWN resistor contacts are closed which pulls the input pin up to 5V which is the state of the microcontroller. In the example below the input pin will return a binary value of 1 which is a higher or pulled to binary code.</p>  <p>In the second diagram, which we can see below, the PULL-DOWN resistor contacts are open and that pulls the input pin down to 0 because in open state the relatively high resistance of the LED also pulls the LED resistor to 0. In the state of the switch where the open contacts of the switch offers infinite resistance, in the example below the input pin will return a binary value of 0 which is a higher or pulled to binary code.</p>   <p>So, in the world of the (most) 8-bit PIC microcontrollers... and other microcontrollers... when an input pin is pulled up to 5V the high voltage registers as a binary 1 and when the input pin is pulled down to 0 that value</p>	<p>PAGE 64 ROBOTICS 101 HANDBOOK 1 ©2018 MICHAEL PETERSHANS</p> <p>registers as a value of binary 0. On the following page there is code that pulls the value of an input pin and displays that value on the serial monitor so that as you press and release the push-button you can see the value.</p> <pre data-bbox="853 1411 1212 1568">int pinNumber = 0; int ledPin(); Serial.begin(9600); void setup() { pinMode(pinNumber, INPUT); } void loop() { int pinValue = digitalRead(pinNumber); Serial.println(pinValue); delay(500); }</pre> <p>The above code will monitor the input state of a switch connected to digital pin D0 and display that value to the serial monitor screen, refreshing every 500 milliseconds.</p> <p>To open the serial monitor screen, you need to click on the magnifying glass icon in the top-right hand corner of the main Arduino IDE screen and you will see the result as the right if the switch is pressed (switch is closed) and the result as 0 if the switch is not pressed (switch is open).</p> 