***Introduction to Robotic Systems Course***

**Getting Started Guide**

**Issue 1.0**

Contents

[1 Introduction 1](#_Toc33695510)

[1.1 Scope of this guide 1](#_Toc33695511)

[2 OpenCR1.0 board setup 2](#_Toc33695512)

[2.1 Software requirements 2](#_Toc33695513)

[2.1.1 Keil MDK 2](#_Toc33695514)

[2.1.2 Arduino IDE 2](#_Toc33695515)

[2.2 Hardware requirements 2](#_Toc33695516)

[2.3 Debugger set up in Keil MDK 3](#_Toc33695517)

[2.4 OpenCR1.0 and Arduino IDE 3](#_Toc33695518)

[2.4.1 Set up the board URLs 3](#_Toc33695519)

[2.4.2 Configure the boards manager, ports and programmer 3](#_Toc33695520)

[3 Remote PC and Raspberry Pi setup 4](#_Toc33695521)

[3.1 Raspberry Pi 3 Setup 4](#_Toc33695522)

[3.2 Remote PC Setup 4](#_Toc33695523)

[4 Troubleshooting 5](#_Toc33695524)

[4.1 Issues when using the Keil MDK 5](#_Toc33695525)

[4.1.1 Cannot download code to the board 5](#_Toc33695526)

[4.2 Issues when using the Arduino IDE 7](#_Toc33695527)

[4.2.1 Error while uploading sketch or failing to jump to boot 7](#_Toc33695528)

[5 Additional references 8](#_Toc33695529)

[5.1 OpenCR Board Manual 8](#_Toc33695530)

[5.2 Robot Operating System ROS 8](#_Toc33695531)

[5.3 Dynamixel Workbench 8](#_Toc33695532)

# Introduction

This guide is intended to give you the information required to install and configure the tools needed to run the labs for the Arm Education: Introduction to Robotic Systems Course. The TurtleBot 3 Burger (a robot) is required to run the labs in this course. The robot has a microcontroller (OpenCR1.0 board) and an on-board computer (Raspberry Pi 3). This document is aimed at guiding you to install the software and environment needed to configure and program the microcontroller and the on-board computer.

## Scope of this guide

This guide is divided into three parts:

* **OpenCR1.0 board setup**
* **Remote PC and Raspberry Pi 3 setup**
* **Troubleshooting**

**OpenCR1.0 board setup** describes:

* How to get started with the OpenCR1.0 board.
* Software and hardware requirements of the OpenCR1.0 board.

**Remote PC and Raspberry Pi 3 setup** describes:

* How to get started with the TurtleBot 3 Burger on board computer (Raspberry Pi 3).
* Software and hardware requirements of the Raspberry Pi 3.
* Install Ubuntu on Virtual Machine.

**Troubleshooting**:

* Provides guidance on how to fix some issues one might encounter with the OpenCR board.

# OpenCR1.0 board setup

This section describes the setup and installation of the software and hardware required to program the OpenCR1.0 board.

## Software requirements

Table 1 shows the list of required software tools to run the labs targeted at the OpenCR1.0 board. The software versions listed here have been verified to work with our labs.

Table 1: Required IDEs

|  |  |  |  |
| --- | --- | --- | --- |
| **Software** | **Website** | **Version**1 | **OS**  |
| Keil MDK | <http://www.keil.com/>  | V5.25.2.0 | Windows OS (to-date) |
| Arduino IDE | <https://www.arduino.cc/>  | 1.8.7 | Windows, Mac OS X, Linux, Linux ARM |

### Keil MDK

The Arm Keil MDK-Lite Edition is a free limited version that provides enough functionality for this course. With this tool, you can edit C, C++, assembly etc codes targeted at the microcontroller and use its debug tool to view the processor state, test, verify and optimize your code.

### Arduino IDE

Arduino IDE is an open source program with which you can write code and upload to the OpenCR board. In the labs for this course, codes written using the Arduino IDE will target peripherals like the Dynamixel Motors, Infra-red sensors, etc attached to the board.

## Hardware requirements

Table 2 shows the list of hardware needed to connect the board for programming. Items 1, 2, 3, 4 in Table 2 are needed to program and debug the OpenCR board when using the Keil MDK.

Table 2: Hardware required

|  |  |  |
| --- | --- | --- |
| **No.** | **Hardware** | **Connections**  |
| 1. | ULINK-ME debugger | Has a JTAG (20-pin) header and a Mini B USB port |
| 2. | Cortex SWD 10-pin Adapter | Has a JTAG (20-pin) port and a 10-pin SWD port |
| 3. | 10-pin 2x5 Socket-Socket SWD Cable | Connects the OpenCR and the Cortex SWD 10-pin Adapter |
| 4. | Micro A to Mini B USB cable | Used to connect the debugger to the computer. |
| 5. | Micro A to Micro B USB cable | Connects OpenCR and computer (Arduino IDE programming) |

Table 3 shows additional list of hardware (sensors) needed to complete these labs.

|  |  |  |
| --- | --- | --- |
| **No.** | **Hardware** | **Description** |
| 1. | Four IR sensors | Required to complete the labs for line sensing. Suggested IR sensor: 5V Infrared Line Track Tracking Tracker Sensor Module. |
| 2. | Microphone | Required to complete the lab for voice control.  |

## Debugger set up in Keil MDK

To program and debug your application code on the OpenCR board, you need to set up the driver for the debugger in the Keil MDK. To set up the ULINK-ME debugger driver, follow the steps below.

1. Connect the ULINK-ME debugger to the microcontroller and your computer.
2. Launch Keil MDK.
3. Select **Projects** **-> Options for Target**.
4. Open the **Debug** tab in the **Options for Target** window.
5. Select the **ULINK2/ME Cortex Debugger** option using the dropdown menu on the right side of the window.

## OpenCR1.0 and Arduino IDE

To program the OpenCR1.0 board using Arduino IDE, connect the board directly to the computer using the Micro A USB port on the board.

If this is a newly installed Arduino IDE, you may need to set up the board URLs and configure the boards manager, ports and programmer before you can program the board using Arduino.

### Set up the board URLs

To set up the board URLs, follow the steps below. Step 3 assumes Arduino is installed in a Windows computer. If using a Linux or Mac computer, choose the correct option in step 3.

1. Launch the Arduino IDE.
2. Select **File** -> **Preferences**.
3. In the “Additional Boards Manager URLs” text box, enter the preferences url found in this link “<http://emanual.robotis.com/docs/en/parts/controller/opencr10/#install-on-windows>”.
	1. On the webpage go to **Arduino IDE -> Install on Windows -> Porting to Arduino IDE(Windows)**” to copy the Preferences url.

### Configure the boards manager, ports and programmer

To configure the boards manager, ports, and programmer in the Arduino IDE, follow these steps:

1. Set the board to OpenCR:
	1. Click **Tools.**
	2. Select **Tools -> OpenCR**. If OpenCR does not exist, then:
		1. Select **Boards Manager**.
		2. Type “OpenCR” in the search box.
		3. In the OpenCR by ROBOTIS option shown, **install version 1.3.2**. This version is verified to work with the labs for this course.
		4. When installation is complete, close the Boards Manager window.
		5. Repeat steps 1a and 1b.
2. Set the port:
	1. Click **Tools**.
3. Select the correct port for OpenCR from **Port -> COMxx** If unsure of the correct port:
	1. Open **Device Manager** on your windows machine.
	2. Expand **Ports**. You should see **OpenCR Virtual ComPort in FS Mode (COMxx)** listed in **Ports**, where COMxx is the port name. If the port is not listed in the **Device Manager**, then check step 5 (a - h) in [Cannot download code to the board](#_Toc30336985) which will guide you on how to install the drivers for **OpenCR Virtual ComPort in FS Mode**.
4. Set the programmer:
	1. Click **Tools**.
	2. Select **Programmer ->** **DFUT\_Util**.

For more information on configuring Arduino, see <http://emanual.robotis.com/docs/en/parts/controller/opencr10/>.

# Remote PC and Raspberry Pi setup

In this section, links to the resources required to set up the environments to control the TurtleBot 3 are provided.

To control the TurtleBot 3, Ubuntu OS should be installed in the Remote PC and a Linux distro based on Raspbian should be installed on the Raspberry Pi 3.

If using a Windows machine, you can run Ubuntu on a Virtual Machine (VM) as the Remote PC.

## Raspberry Pi 3 Setup

Follow the instructions in Install Linux (Raspbian) option in the link below.

<http://emanual.robotis.com/docs/en/platform/turtlebot3/raspberry_pi_3_setup/#raspberry-pi-3-setup>

## Remote PC Setup

Download and install Ubuntu 16.04 from <http://releases.ubuntu.com/16.04/>. The labs in this course were tested with this release.

You might want to install Ubuntu on a virtual machine (e.g. VirtualBox which can be downloaded from <https://www.virtualbox.org/wiki/Downloads>) for convenience. Ensure to allocate at least 25GB of storage when setting up your virtual machine.

# Troubleshooting

This section provides some guidance on resolving known issues when using the OpenCR1.0 board with the Keil MDK and Arduino IDEs.

## Issues when using the Keil MDK

### Cannot download code to the board

To resolve this issue, follow these steps:

1. Ensure that the board is connected to the computer through the debugger.
2. Ensure that the correct driver for the debugger is selected. To do this in the Keil MDK:
	1. Select **Projects -> Options for Target**.
	2. Open the **Debug** tab in the **Options for Target** window.
	3. On the right side of the **Debug** tab, select the driver for the debugger on the **Use** drop-down menu. For ULINK-ME debugger, **select** the **ULINK2/ME Cortex Debugger** option.
3. Next click the **Settings** button next to the drop-down menu, as highlighted in Figure 1.



Figure 1: Debugger options in Keil MDK

1. Ensure that the IDCODE and Device Name appear under the **SW Device**, as shown in Figure 2 if it is not shown, skip to step 5 first and if it does not work, do step 6. Note, the IDCODE and Device Name may not be the same as shown in Figure 2.



Figure 2: SW Device IDCODE and Device Name in Keil MDK.

1. No IDCODE and Device Name under **SW Device** in the Keil MDK as shown in Figure 2? Follow these steps:
	1. Connect the OpenCR board to the computer using the micro-USB port on the board.
	2. Press and release the Reset button on the OpenCR1.0 board.
	3. Download and install Zadig from <https://zadig.akeo.ie/>. This is a Windows application that enables you to install the correct drivers for the USB devices
	4. Open Zadig and click on **Options** and select **List All Devices**.
	5. Select **OpenCR Virtual ComPort in FS Mode** in the drop-down menu.
	6. Select the **USB Serial (CDC)** option for the **Driver**.
	7. Click **Replace Driver**.
	8. Remove and reinsert the USB cable on the computer and go through steps explained in [Cannot download code to the board](#_Cannot_download_code).
2. Try this step only if step 5 failed and the IDCODE and Device Name under **SW Device** does not show.
	1. Connect the OpenCR board to the computer using the micro USB port on the board.
	2. On the OpenCR board, hold down the BOOT0(DFU) button and press and release the Reset button. Then release the BOOT0(DFU) button. The board is now in BOOTLOADER mode. For information on the board layout, see <http://emanual.robotis.com/docs/en/parts/controller/opencr10/#run-dfu-mode-1>.
	3. Ensure that Additional Boards Manager URLs, correct Board, port and Programmer options are selected.
	4. In the Arduino IDE, select **Tools -> Burn Bootloader**.
	5. If it is successful, you will see the *File downloaded successfully* message displayed on the bottom of the Arduino IDE.
	6. Press and release the Reset button on the board to exit the BOOTLOADER mode.
	7. Remove and reinsert the USB cable on the computer and go through the steps described in [Cannot download code to the board](#_Cannot_download_code), repeat the initial step 5 if needed.

## Issues when using the Arduino IDE

### Error while uploading sketch or failing to jump to boot

This section provides troubleshooting recommendation if you encounter this error message:

An error occurred while uploading sketch

or, when you try to upload sketch, this occurs:

Fail to open port 1 : COMx Fail to jump to boot

To troubleshoot the issues stated in the error messages above, follow these steps:

1. Install OpenCR 1.0 board version 1.3.2 on the Arduino IDE **Boards Manager** in the **Tools** options.
2. Ensure that the OpenCR board is connected to the computer using the micro-USB port on the board.
3. Ensure that the OpenCR board is connected to a power source and the power switch turned ON (especially if the sketch is to control the motors).
4. Enter bootloader state.
5. Download and install Zadig from <https://zadig.akeo.ie/>. This is a Windows application that enables you to install the correct drivers for the USB devices. You may skip this step if you already have Zadig on your computer.
6. Open Zadig and select **STM32 BOOTLOADER** in the drop-down menu.
7. Select the **WinUSB (v6.1.xxxx.xxxxx)** option for the **Driver**.
8. Click on **Replace Driver**.
9. In the Arduino IDE, select **Tools -> Burn Bootloader**.
10. Press and release the Reset button on the OpenCR1.0 board. You can try to upload the sketch now.
11. Open Zadig again, click **Options** and select **List All Devices**.
12. Select **OpenCR Virtual ComPort in FS Mode** in the dropdown menu.
13. Select the **USB Serial (CDC)** option for the **Driver**.
14. Click on **Replace Driver**.

# Additional references

## OpenCR Board Manual

<http://emanual.robotis.com/docs/en/parts/controller/opencr10/>

## Robot Operating System ROS

<http://wiki.ros.org/>

## Dynamixel Workbench

<http://emanual.robotis.com/docs/en/software/dynamixel/dynamixel_workbench/> and <http://wiki.ros.org/dynamixel_workbench_controllers>