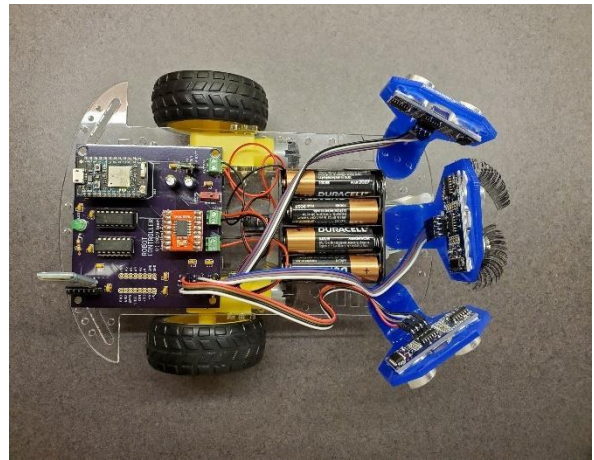


# Robot Project Overview

By: Jim Schrempp and Bob Glicksman; v2, 10/8/2020

**NOTICE:** Use of this document is subject to the terms of use described in the document “Terms\_of\_Use\_License\_and\_Disclaimer” that is included in this release package. This document can be found at:

[https://github.com/TeamPracticalProjects/Robotics/blob/master/Terms\\_of\\_Use\\_License\\_and\\_Disclaimer.pdf](https://github.com/TeamPracticalProjects/Robotics/blob/master/Terms_of_Use_License_and_Disclaimer.pdf)



© 2020 Team Practical Projects, Bob Glicksman and Jim Schrempp. All rights reserved.

# **TABLE OF CONTENTS.**

TABLE OF CONTENTS.	1
1. DOCUMENT OVERVIEW.	2
2. REPOSITORY OVERVIEW.	2
Repository Structure.	2
Suggested Use of the Documents.	4
3. ROBOT PROJECT ARCHITECTURE AND COMPONENTS.	5

# 1.DOCUMENT OVERVIEW.

This document provides an overview of the Robot project. The Team Practical Projects Robot is designed for learning and entertainment. You can control the Robot manually, via an Android App, or have it navigate autonomously, sensing and avoiding obstacles as it goes. The Robot is based upon the Particle<sup>1</sup> Photon; an inexpensive, but advanced, 32 bit microcontroller with built-in WiFi capability. The Photon's WiFi capability is used in this project to allow new firmware to be flashed to the Robot "over-the-air", without the need for wired connections. The Robot itself has been designed for Bluetooth control, so that it can operate anywhere, regardless of whether a WiFi connection is present or not.

Section 2 of this document provides a guide to this Repository. The Robot is an open source project and we provide complete documentation, including source code for the Robot firmware and controlling App software, and CAD files for the printed circuit board (PCB) that holds the Robot's electronics. We also provide a complete build and installation instruction manual and a user manual.

Section 3 of the document provides an overview of the system architecture. The "system" is very simple, consisting of the Robot hardware, the Robot's internal firmware for the Photon microcontroller, and an Android App that is used to control and monitor the Robot.

# 2.REPOSITORY OVERVIEW.

This section describes the contents of this repository. This is an open source project. The files in this repository contain source code, hardware schematics and CAD files, and documentation.

## Repository Structure.

### GitHub.com/TeamPracticalProjects/Robotics

- | -- "*README.md*": the overview read-me file that is displayed on the GitHub project page. The file format is *markdown (.md)*
- | -- "*Terms\_of\_Use\_License\_and\_Disclaimer*": the terms of use that you must agree to in order to use the contents of this repository. The file format is *portable document format (.pdf)*
- | -- "*Robot\_Project\_Overview.pdf*": This document. The file format is *portable document format (.pdf)*
- | -- "*Robot.png*": a photo of the Robot. The file format is *.png*
- | -- **Software** (folder containing all software and firmware):
  - | -- **AI2\_App** (folder containing the app software)
  - | -- **compiled** (folder)
  - | -- "*BasicRobotControl.apk*": Android install package

---

<sup>1</sup> <https://www.particle.io/>

for the app. Load this file onto your Android smartphone and tap on it to install the app. The file format is *android application package (.apk)*

| -- **src** (folder)

| -- “*BasicRobotControl.aia*”: Source code for the app.

The source code is in the MIT App Inventor 2 graphical language. Import this file into your MIT App Inventor 2 IDE in order to view and/or edit the app source code. The file format is *A/2 source code format (.aia)*.

| -- **Firmware** (folder containing the firmware for the Photon)

| -- “.gitignore”. This file is used to ignore some files that should not be added to your repository.~~This is automatically added by GitHub and you do not~~

~~need to do anything with this file.~~ It is in *plain text format*.

| -- “*AutonomousRobotParticle.code-workspace*”. This file is automatically added by the Particle Workbench IDE and you do not need to do anything with it. If you use the Particle Workbench you can open this file to restore a development environment. It is in *plain text format*.

| -- “*project.properties*”. This file is automatically added by the Particle Workbench IDE and you do not need to do anything with it. It is in *plain text format*.

| -- **src** (folder)

| -- “*AutonomousRobotParticle.ino*”: Particle Photon source code for the firmware that is loaded onto the Photon. The file format is *Arduino source code format (.ino)*, which is pure text. Copy the contents of this file into the Particle Web IDE or the Particle Workbench in order to compile the firmware and flash it onto your Photon.

| -- **Documents** (folder containing project documentation)

| -- “*Robot\_Build\_And\_Installation\_Instructions.pdf*”: Document that details how to build, install and test the project. The file format is *portable document format (.pdf)*.

| -- “*MFG\_robot\_assembly\_instructions.pdf*”: The robot chassis manufacturer’s instruction sheet for how to assemble the robot chassis kit. The file format is *portable document format (.pdf)*.

| -- “*Robot\_User\_Manual*”: User manual for the Robot and its associated App. The file format is *portable document format (.pdf)*.

| -- **Hardware** (folder containing project hardware files)

| -- “*Robot\_Parts\_List.pdf*”: A parts list for the Robot with links to sources for the parts. The file format is *portable document format (.pdf)*.

| -- “*Photon\_Robot\_View\_1.jpg*”: A high resolution, top view photo of the Completed Robot. The file format is *JPEG (.jpg)*.

- | -- “*Photon\_Robot\_View\_2.jpg*”: A high resolution, side view photo of the Completed Robot. The file format is *JPEG (.jpg)*.
- | -- “*Caster\_mounted\_with\_standoffs.jpg*”: A photo showing how the ball caster is attached to the Robot chassis. The file format is *JPEG (.jpg)*.
- | -- **Photon PCB** (folder)
  - | -- **Eagle\_Files** (folder)
    - | -- “*RobotController.sch*”: Cadsoft Eagle schematic. File format is *Eagle schematic (.sch)*.
    - | -- “*RobotController.brd*”: Cadsoft Eagle board layout. File format is *Eagle board (.brd)*.
  - | -- “*Robot\_Controller\_PCB\_Parts\_List.pdf*”: A parts list for the Robot PCB. The assembled PCB is itself a part on the *Robot\_Parts\_List* in the **Documents** folder. The file format is *portable document format (.pdf)*.
  - | -- “*Robot\_Controller\_Schematic.pdf*”: This is a PDF version of The Robot PCB schematic, for those who don’t have Eagle. The file format is *portable document format (.pdf)*.

## Suggested Use of the Documents.

We suggest that you read the documents (.pdf files) contained in this repository in the following order:

- *Terms of Use License and Disclaimer*: read and agree to everything in this document before proceeding further this this project.
- *Robot Project Overview*: read through this document to familiarize yourself with what this project is about and ensure that you can successfully complete, install and use this project before you invest further in it.
- *Robot Build And Installation Instructions*: follow the steps in this document to build the electronics, build the Robot chassis and install its components, install the Photon firmware, install the App software, and perform a basic test that things are working.
- *Robot User Manual*: consult this manual to learn how about the switches on the Robot and learn to use the App.

### **3.ROBOT PROJECT ARCHITECTURE AND COMPONENTS.**

The overall architecture of the Robot system is shown in figure 3-1. The major components of the system are:

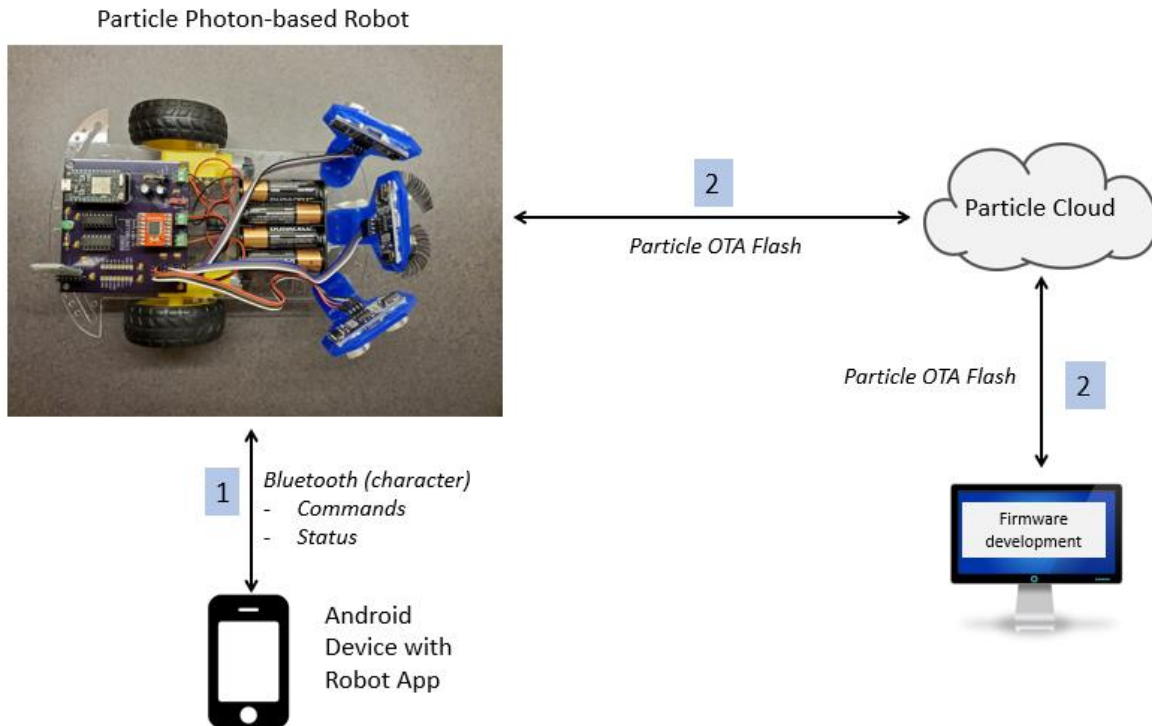
- The Robot itself.
- An Android<sup>2</sup> device (phone or tablet) running the Robot control app.
- A development computer (desktop, laptop, tablet) for compiling the Robot firmware source code and flashing the resulting firmware onto the Robot's Photon module over the Internet.

This repository contains Particle source code for the Robot. Particle's source code is written in Particle's firmware language which is Arduino compatible and compiled with a C++ compiler. This source code can be compiled for the Photon using either Particle's Web IDE or Particle's Workbench IDE. After compilation, either of these IDE's can be used to flash the resulting executable firmware onto the Photon using Particle's over-the-air (OTA) flashing capability via the Particle Cloud. Instructions for doing this are contained in the document "*Robot\_Build\_And\_Installation\_Instructions.pdf*" which can be found in the "**Documents**" folder in this repository. Note that the Photon uses WiFi to communicate with the Particle cloud over the Internet, so flashing of new firmware onto the Robot can only take place where the Photon has access to WiFi and where the Photon is actually on the WiFi network. Particle makes this process extremely easy and their documentation is excellent (see: <https://docs.particle.io/photon/>).

After firmware is flashed to the Photon, the Robot no longer has to be in WiFi range in order to operate. The Robot is battery powered and communicates with the App using Bluetooth. The firmware places the Robot's Photon in "Semi-Automatic mode" which means that the firmware does not need to be connected to the Particle Cloud in order to operate. However, when you want to flash new firmware the Photon will need to be connected to WiFi; do this by placing the Photon into "safe mode". Instructions for doing this are contained in the document "*Robot\_Build\_And\_Installation\_Instructions.pdf*" which can be found in the "Documents" folder in this repository.

---

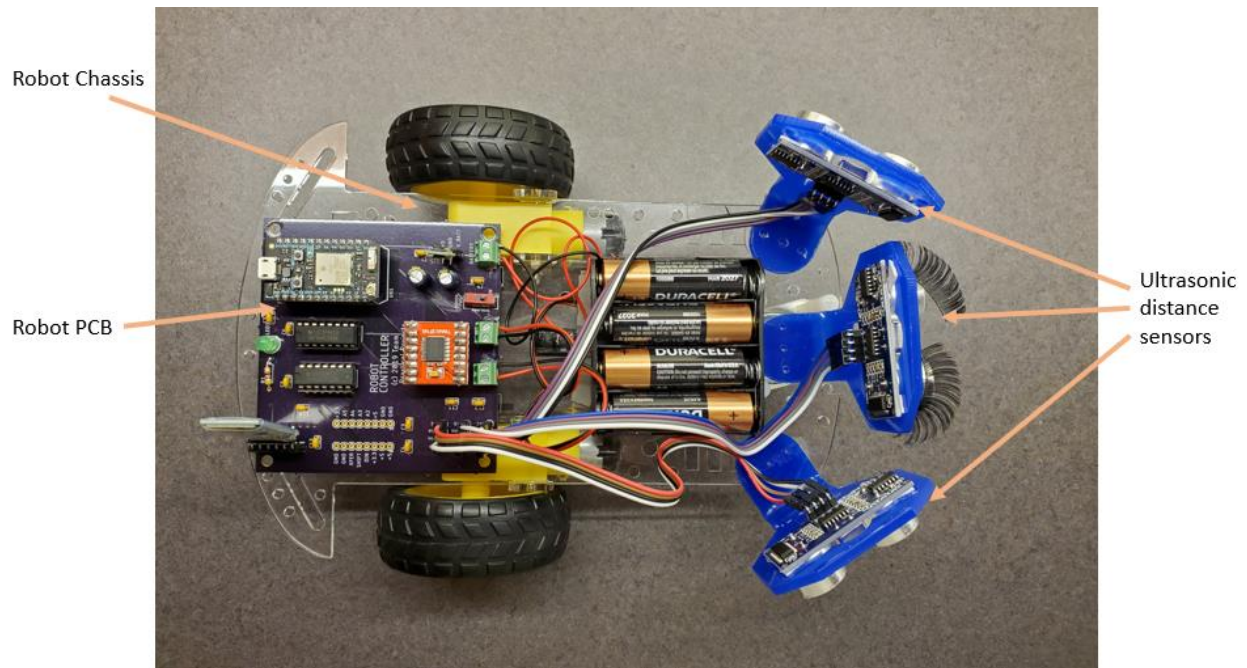
<sup>2</sup> Sorry; an iOS version of the Robot app is not available at this time.



*Figure 3-1. Robot System Components.*

Figure 3-2 is a closeup photo of the Robot. The Robot consists of three major components:

- The Robot Chassis. This is a low cost kit that you purchase over the Internet. It contains the Robot body and mounting hardware, two motors with encoders and wheels, a front caster wheel (which is not used; we replace this with a ball caster for better performance), a 4 – AA battery holder, and a power on/off switch.
- The Robot PCB. This printed circuit board (PCB) needs to be assembled based upon the parts list in the **Hardware/Photon PCB** folder in this repository.
- Ultrasonic Distance Sensors: The Robot uses three HC-SR04 ultrasonic distance sensors for autonomous navigation. The sensors and their mounts are included in the “Robot\_Parts\_List.pdf” document in the **Hardware** folder in this repository.



*Figure 3-2. Robot closeup.*

The final component of this Robot System is an app that runs on an Android device (phone or tablet). A screen shot of the App (in AUTO mode) is given in figure 3-3. Instructions for using the App are contained in the document “*Robot\_User\_MauaI.pdf*” which is in the **Documents** folder in this repository.



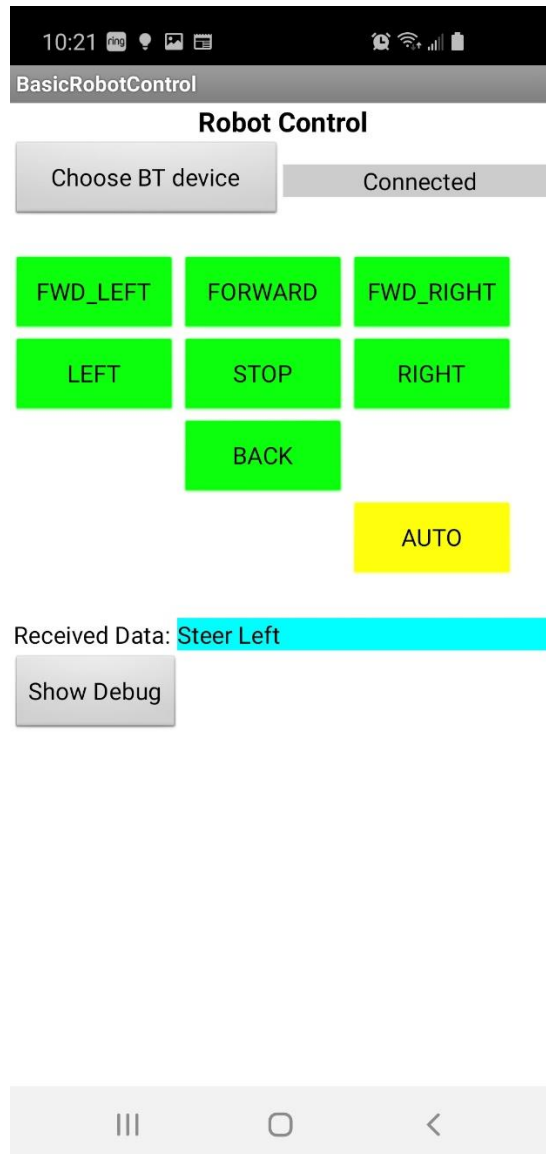


Figure 3-3. App screen shot (AUTO mode).