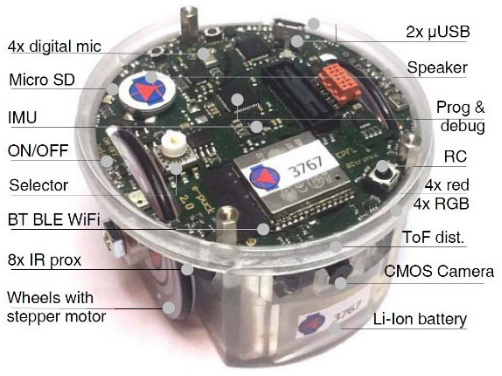
e-puck2 Robot Primer



e-puck education robot (gctronic.com)

Welcome to programming the e-puck robot! This primer document should give you all the help you need to get started.

The e-puck2 robot is a small, autonomous robot used for teaching and hobbyists. As in the inset, it has several sensors and actuators, including IR sensors, a time of flight distance sensor, accelerometer, lights, and two stepper motor wheels, etc. The epuck2 is an open platform that is fully documented and has a developer community. Check out their website.

Of particular interest is the 5Wh battery (about 2-3 hours of heavy use) and radio (for Bluetooth or wifi), which means the robot can work untethered. Further, the robot has a small DSP microcontroller with an FPU (STM32F4 at 168MHz, 210 DMIPS) for signal processing.

The robot has a programmable firmware that you can update and place your code on, to make the robot fully autonomous and standalone. However, for the purposes of this course we will not be programing this firmware (that would be more of an embedded systems course) and will keep the stock installed firmware for the main robot operation.

The robot can be charged over USB (there are two ports, equivalent I think). There is a blue button on the bottom that you can use to turn it on or off (long press). There is a selector switch on the top of the robot to tell it which internal program to run when it re-sets.

Programming the Robot

Instead of writing your own firmware and dealing with the on-board computer, you will instead be working at a higher level, focused more on robot behavior programming. To achieve this, you will be connecting to the e-puck2 by a wireless connection (Bluetooth and/or Wifi) and sending commands over the air to direct the robot and read sensor values.

What is the difference? The Bluetooth is more flexible and easy to connect to from various sources but uses a somewhat esoteric connection mechanism. The WiFi has a higher bandwidth (higher video frame rate) and should have a more reliable connection, but requires more overhead to get setup.

Programming the Robot Firmware

You may never have to do this. However, you can try this if a) your robot stops behavior properly and you suspect a corrupted firmware (I've never had this happen), or b) you want to connect to your robot using WiFi.

These instructions are simplified from those provided by the manufacturer: e-puck2 - GCtronic wiki. These instructions are for windows PCs, see the manufacturer site for linux or Mac instructions and software.

We have provided a firmware .zip file with a readme explaining the different files. These are all from the GCtronic website.

Programming the Main Robot Firmware

You generally will not do this but only as a recovery mechanism. Simply use the main programmer software with the provided .elf file. Connect the robot, open a cmd window and run the “program.bat” command. It will end in GDB without much of a “success” message, but you won’t get an error.

Programming the Radio Firmware

You will do this to switch the robot radio between connecting to Bluetooth or to WiFi capability. Actually establishing the connection is a different problem outlined below. Unzip the radio programmer, and further unzip the binary firmware for the version you want (wifi or Bluetooth) into the same folder. Run the “program.bat” file and it will do a hard reset once done.

Connecting to the Robot

Establishing your first connection to your robot can be quite the challenge as it requires you to setup a channel, have the correct address, have the robot configured correctly, have your development environment setup correctly, and then run your code.

The sample code and provided library for this class are all in Python 3. When running the library it may have dependencies that you do not have installed so be prepared to use your favorite package manager (e.g., pip) to install them. It’s your job to setup your environment and your favorite IDE, e.g., VSCode works fine.

Connecting to the Robot over Bluetooth

Your robot comes programmed with a Bluetooth firmware. If not, see the earlier section on how to program it. Once you create a Bluetooth connection, the driver creates virtual Communications Ports, or COM ports. These are emulations of a very very old system that we used to use for serial communications with devices. They are also often called serial ports.

Set the robot’s physical selector switch (on the top) to 3, and reset the robot.

If the robot is not already paired with your device you need to pair it. To put the robot into pairing mode, hold the esp32 button (top, near the rear) while you power it on to enter pairing mode. It should ask you to confirm the number that matches the robot’s printed sticker. If asked for a pin try 0000 or this number.

You need to find the COM port (of the format COMXXX where XXX is a number, e.g., COM3, in windows/dos eco-systems) that connects to the robot’s protocol end point. There will be several so be careful to choose the right one (typically labelled UART). GCTronic provides explanations of how to find your endpoint for various Oss: e-puck2 PC side development - GCtronic wiki. This becomes your address to connect to the robot.

Once you have this, download and run our HelloWorld program, uncomment the “serial” connection lines, and enter your address. If all goes well your robot will flash lights, make noise, and turn in circles. If it doesn’t, make sure you have the correct end point.

Connecting to the Robot over WiFi

You will need to reprogram your robot’s radio firmware for WiFi before this works. See above.

Set the robot’s physical selector switch (on the top) to 15/F, and reset the robot.

See the full instructions on the GCTronic website. A quick summary is that if this is the first time you setup WiFi, it will be in access point mode and you can connect directly to it to configure it. You then tell it which WiFi network to connect to, and it will automatically connect to that network, refresh, and tell you the device's IP. You need this IP to connect to the robot from a device on the same local network.

Unfortunately the robot's WiFi code is very primitive and cannot do much authentication. It is possible that it will not connect to your home network, and it certainly will not connect to uofm-secure. If you want to work with WiFi on campus, you need to use a different approach such as using your laptop or phone as an access point.

We have had excellent success on android phones with setting up a simple access point for the robot and a laptop to connect to, and using that network to communicate with the robot.

Once you have this, download and run our HelloWorld program, uncomment the "WiFi" connection lines, and enter the robot's IP address. If all goes well your robot will flash lights, make noise, and turn in circles. If it doesn't, make sure you have the correct IP and are on the same network.