

Explorer Bot resources

1. Arduino

Task 2 consisted of hardware testing and interfacing. **All the teams are allowed to use any Arduino libraries.** Arduino has lots of libraries and examples codes which can be programmed using Arduino IDE.

2. Multitasking with Arduino library

Designing solution of system depends on the requirements. Some systems consist simple single task, while some consist of multiple complete task. Also some tasks are time dependent and some are independent. So how to design solution for such systems. If we know how to write solution for multitask system, we can program single task system as well. As single task system is just subset of multi task system with only one sub task. Further idea on creating multitasking solutions is available in *Multitasking with Arduino* folder. These tutorials, provided by Adafruit in three parts, will give you clear picture of using timer as `milli()` function for multitasking.

3. Implementation of PID algorithm in Embedded system

PID algorithm is a robust algorithm for error correction. It is one of the robust control algorithm for correction of error in a dynamic system. Tutorial on implementation of PID control is available on following [link](#). These tutorial provides the practical implementation of PID algorithm. It discusses how to go from continuous time to discrete time using Sample time. It also discusses other problems like Derivative kick, Reset Windup, Controller Direction which arise in the implementation of PID algorithm in practice.

Above tutorial provides only the implementation of PID algorithm but does not say anything about PID tuning. For tuning, various methods such as Ziegler-Nichols method are available but it requires full system simulation model, which may not be possible for Balance Bot. So simplified version of above method is widely used which does not require system model. In simplified version, first parameter for Proportional Control is changed keeping Integral and Derivative parameters to zero. Then after getting optimum output Derivative parameter is changed and finally to make error zero Integral term is changed.

4. Interfacing Quadrature Encoder

Quadrature encoder is a position sensor which gives angular position of shaft in rotational motion. There are two types of encoders - Incremental and Absolute. Incremental encoder gives position relative to initial position at start of the rotational motion whereas Absolute encoder gives angular position irrespective of shaft position. Quadrature encoder belongs to incremental encoder category. Simple incremental encoder gives single square wave pulse

during shaft rotation. So knowing distance per pulse, resolution we can calculate angular position. But quadrature encoder gives two square waves with phase difference of 90 depending upon direction of rotation.

There is various method for getting position from quadrature. One is based on state machine whose tutorial is available at following [link](#). Another method is implementation based on look up table whose example is available at following [link](#).

5. ROS Interfacing:

Follow the book **Learning Robotics Using Python – Lentin joseph** to interface Explorer bot with ROS.

