PSoC / Zumo group assigments

Week 2

Assignment 3 requires Zumo robot. Other assignments work also with the plain processor board. If you get stuck with the assignments ask the instructor or your peers for help.

Assignment 1

Write a program that blinks S-O-S morse code when uses presses the button. The code is: dot, dot, dot, dash, dash, dash, dot, dot, dot, (long pause if code repeats). The length of dash is three times the length of a dot. The off time between dots and dashes is one dot long. For example, if dot lenght is 0.5 seconds the sequence is 0.5s ON, 0,5 s OFF, 0,5 ON, 0,5 s OFF, 0.5 s ON, 0.5 s OFF, 1.5 ON, 0.5 s OFF, 1.5 s ON, …

After the sequence the program goes back to wait for another button press to start S-O-S again.

Assignment 2

Write a program that asks user to enter his/her age. The program must measure the time it takes for the user to enter the answer. The program must reply according to the following table:

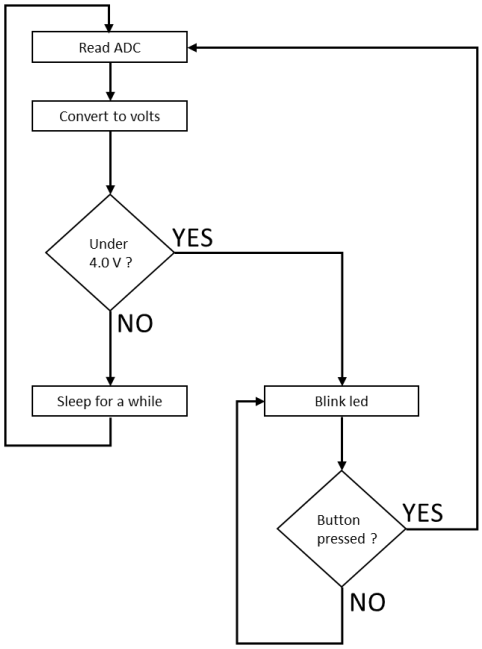
Age / Response time	Less than 3 seconds	3 – 5 seconds	Over 5 seconds
21 or under	Super fast dude!	So mediocre.	My granny is faster than you!
22 – 50	Be quick or be dead	You're so average.	Have you been smoking something illegal?
Over 50	Still going strong	You are doing ok for your age.	Do they still allow you to drive?

You can use C standard library functions for input and output. Note that you must run Putty to see the output and to enter input.

The time can be measured by taking timestamps after the question is printed and when user has entered the answer. The time response time is last timestamp – first time stamp. You get the timestamp by calling xTaskGetTickCount(). The return timestamp type is TickType_t. You need two variables of this type to calculate the response time. The resolution of timestamp is millisecond so your calculation result will give the response time in milliseconds.

Assignment 3

Write a program that measures the battery voltage of the robot, outputs it to the terminal, and alerts user by blinking the on board led if battery voltage is below 4.0 V. User can acknowledge the alert by pressing the button. If the voltage is high enough the blinking stops when button is pressed.

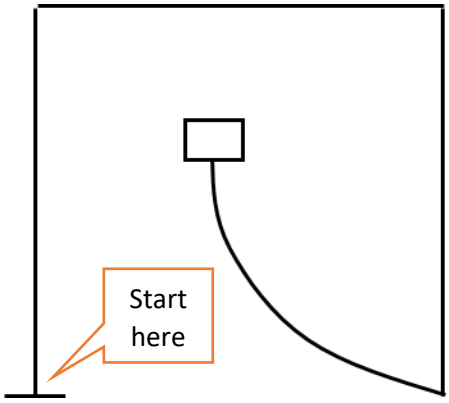


Robot must be powered from batteries in order to measure the voltage. You can easily test your program by switching the battery power off. When battery is switched off the voltage drops quickly below 4.0 V. If the processor board is connected to a USB port the CPU will powered by USB and your program keeps running and detects the voltage drop. Switching battery back on will bring the measured voltage back to normal level and you can acknowledge the alarm.

Week 3

Assignment 1

Robot control without sensors. Make the robot run the track that is taped to class room floor. Robot must wait for user button to be pressed before it starts running. Start from “T” and try to hit the box as close as possible. You may need to balance the motors for example, make one motor run slightly faster than the other, to make the robot run straight.



Assignment 2

Robot control with ultrasonic distance sensor. Write a program that runs forward and checks periodically the distance sensor reading. If an obstacle is closer than 10 cm robot beeps, reverses, and turns left while reversing. Then robot continues to run forward and check for obstacles again.

Assignment 3

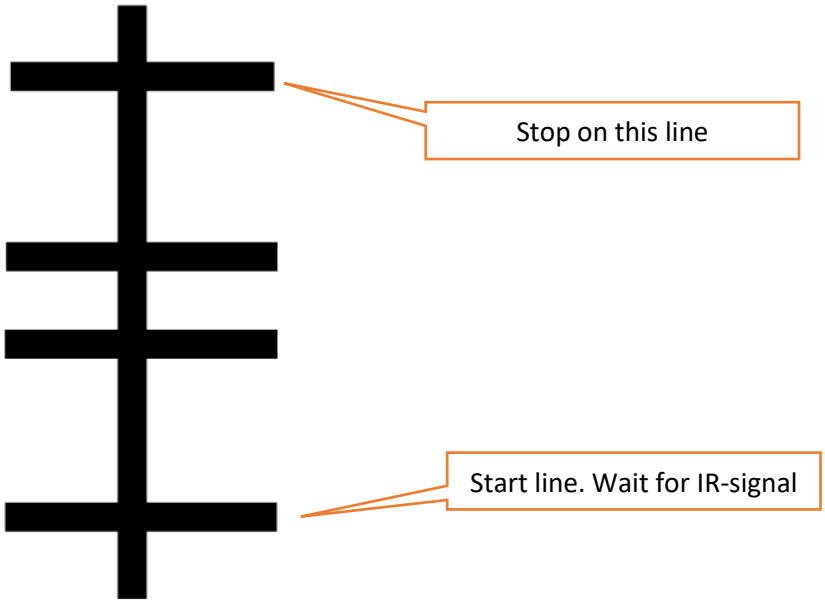
Robot control with accelerometer. Write a program that runs robot forward and makes random turns. Check the accelerometer value periodically. If the robot hits an obstacle make the robot reverse, make a random turn 90 degrees left or right, and start forward again.

Start with the accelerometer example and test the magnitude of sensor values by bumping the robot from different sides and angles to find sensor value threshold for hit/not hit.

Week 4

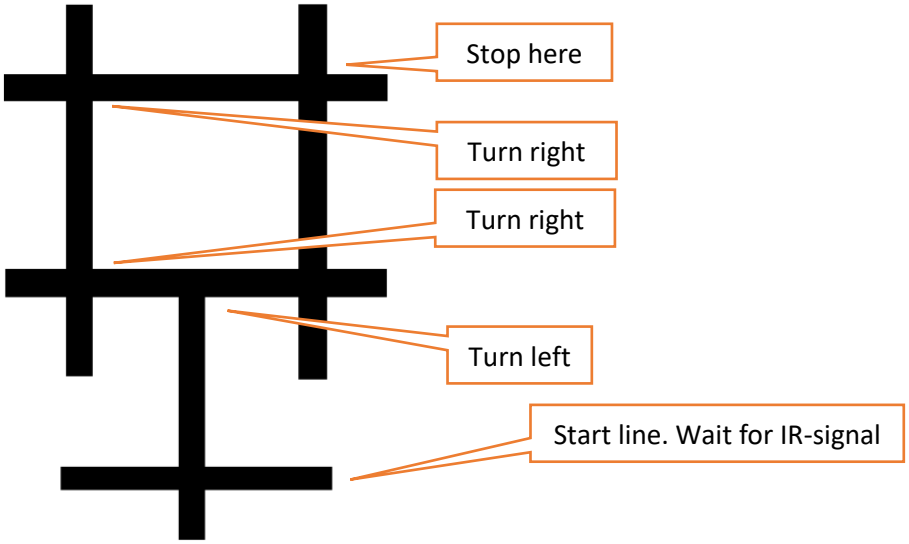
Assignment 1

When user button, the button in the centre of PSoC board, is pressed the robot drives to the first line. On the first line the robot waits for a command from IR-remote. When any IR command is received the robot starts to run forward and stops on the fourth line.



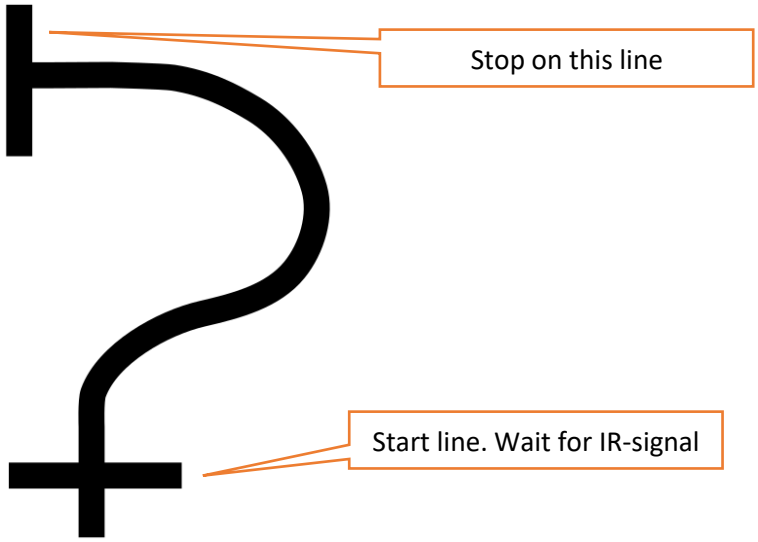
Assignment 2

When user button is pressed the robot drives to the first line. On the first line the robot waits for a command from IR-remote. When any IR command is received the robot starts to run forward. When the robot comes to first intersection it turns left and then takes the next two intersections to the right. The robot stops on the fourth intersection.



Assignment 3

When user button is pressed the robot drives to the first line. On the first line the robot waits for a command from IR-remote. When any IR command is received the robot starts to run forward following the line and stops on the first line.



Week 5

The following assignments require that you have a WiFi network set up, an MQTT broker, and a client that can subscribe messages that your robot sends.

Assignment 1

Write a program that asks the user to enter the current time (hours and minutes). When the user has entered the time, the program enters an infinite loop and sends the current time over MQTT every time when the user presses the PSoC user button.

Use real time clock for time keeping. See Zumo library documentation and examples for details.

Assignment 2

Write a program that uses ultrasonic sensor to detect obstacles. When an obstacle is detected the robot reverses and makes a random 90 degree turn to left or right. The turn direction (left/right) is sent to topic <robot serial number>/turn.

Assignment 3

Write a program that runs the robot to an intersection. When the intersection is reached the robot waits for IR-remote command. After receiving the IR-command the robot runs forward until it sees another intersection. The robot stops at the intersection and sends the elapsed time (from IR-command to next intersection) to topic <robot serial number>/lap.