Project

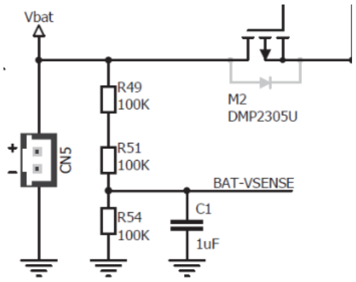
Firmware requirements

- Main loop requirements
  - The control loop should be implemented at 500 Hz frequency.
  - The motor PWM should be updated at 500 Hz and the IR sensor should be read at 500 Hz frequency.
  - The LED A0 and should blink at 1 Hz frequency at all times, to indicate the functioning of the main loop.
  - Initially, the robot should be in “Wait for start” state.
    - In this state, the PWM DC of all the motors should be 0.
  - Once the button RE8 is pressed, the robot should go in the “Moving” state.
    - In this state, the PWM is generated to as it follows
      - Given the speed and yawrate signals received from the UART (see below), the PWM for the left and right motors should be generated accordingly.
    - If the button RE8 is pressed for a second time, the robot should go back in the “Wait for start” state.
  - If the robot, while in the moving state, senses an obstacle closer than a predefined threshold in cm, it should change to a state “Emergency”.
    - In this state, the robot should stop and should send a message $MEMRG,1*. When the obstacle is not sensed for at least 5 second (i.e. the distance is bigger than the threshold, the robot should transition back to the “wait for start” state and should send $MEMRG,0* to signal the end of the emergency stop.
    - The left and right-side lights should blink at 1 Hz to indicate emergency.
    - Pressing button RE8 does not do anything while in this state.
  - In all states, the robot should acquire the filtered accelerometer values x,y,z at 10 Hz.

- Motor control
  - Four PWM signals must be generated to control the buggy, on pins RD1 to RD4 (PR65 to RP68) using four Output Compare peripherals.
  - The frequency of the PWM signals must be 10 kHz.
  - The actuation of the wheels follows the specification reported in the table below:

Command	**PWM with DC > 0**	**PWM with DC = 0**
Left wheels forward (left_pwm > 0)	RD2 = left_pwm	RD1 = 0
Left wheels backward (left_pwm < 0)	RD1 = -left_pwm	RD2 = 0
Right wheels forward (right_pwm > 0)	RD4 = right_pwm	RD3 = 0
Right wheels backward (right_pwm < 0)	RD3 = -right_pwm	RD4 = 0

- Battery sensing
  - The voltage of the battery (BAT-VSENSE in the figure below) is available on pin AN11. It is sensed after a partitioning circuit, i.e., in between a 200 kohm resistor and a 100 kohm resistor.



- IR sensor
  - The infrared sensor should be mounted on the *Buggy Mikrobus* 1 or 2 (i.e., in front of the buggy). The signal can be read on AN14/AN15, while the enable to the IR sensor must be given on the digital I/O on RB9/RA3.
- Data logging / command interface through UART
  - The UART to RS232 module should be installed on the *Clicker Mikrobus* 2. The TX signal should be remapped to RD0/RP64, while the RX signal should be remapped to RD11/RPI75.
  - The microcontroller should send, to the PC, the following messages (in all the states)
    - $MBATT,v_batt* where v_batt is the sensed battery in Volt, at 1 Hz frequency. Use two digits, i.e., X.YZ
    - $MDIST,distance* where distance is the sensed distance in cm, at 10 Hz frequency. Use an integer.
    - $MACC,x,y,z* where x,y,z are the three filtered compoments of the acceleration, in mg, at 10 Hz frequency. Use integers.
  - The microcontroller should receive, from the PC, the following messages (in all the states)

        - $PCREF,speed,yawrate*, where speed is a signal from -100 to 100% for the forward motion and yawrate is also a signal from -100 to 100% indicating the angular velocity (positive being anticlockwise). No acknowledgment is generated.
        - $PCSTP,*, signaling the robot to switch to the wait for start state. The robot should acknowledge the execution of the command with the message $MACK,1*. If the message is received during the emergency state, it no switch should be performed and the robot should acknowledge the PC with the message $MACK,0*.
        - $PCSTT,*, signaling the robot to switch to the moving state. The robot should acknowledge the execution of the command with the message $MACK,1*. If the message is received during the emergency state, it no switch should be performed and the robot should acknowledge the PC with the message $MACK,0*.

    - Given the chosen UART baud rate, the firmware should never lose a message due to its implementation (i.e., proper dimensioning of buffers), even with full use of the bandwidth.

Evaluation criteria

Among other things, these criteria will be used:

- Adherence to the provided specifications
- Correctness of the interrupts service routines
- Correct handling of shared data
- Management of the UART FIFO and circular buffers on both sending and receiving
- General code cleanliness

Pin Mapping

- RB8 Left side lights
- RF1 Right-side lights
- RF0 brakes
- RG1 low intensity lights
- RA7 beam headlights
- AN11 battery sensing
- RD1/RP65 left PWM backward motion
- RD2/RP66 left PWM forward motion
- RD3/RP67 right PWM backward motion
- RD4/RP68 right PWM forward motion
- AN14 or AN15 IR sensor voltage
- RB9 or RA3 IR sensor enable
- RD0/RP64 UART TX
- RD11/RPI75 UART RX