Comp3402 Subsumption Robot

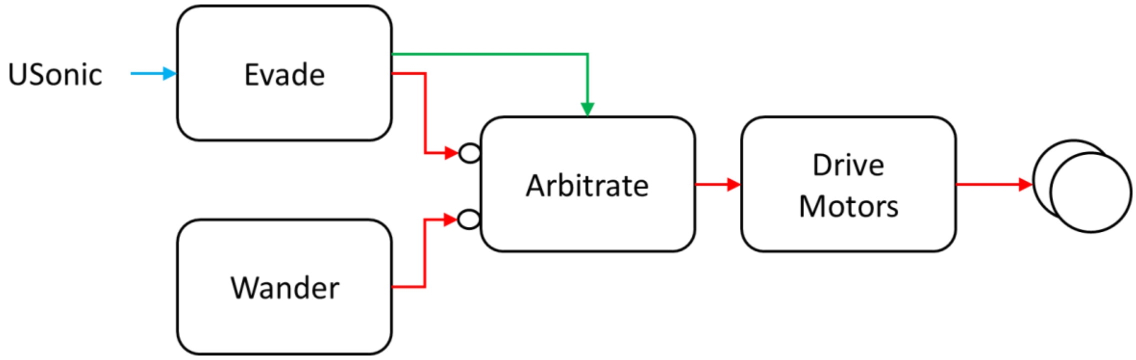
C.B.Price March 2022

Purpose	To create a subsumption robot with Wander and Evade behaviours using the RTOS
Files Required	Arduino **portable** sketchbook
ILO Contribution	6
Send to Me	
Assignment Info.	
Homework	Read Chapter 8

Activities

Architecture

The diagram below shows where we are heading to, the overall architecture of the code. Each rectangle is an individual state. Red lines show the values of the motor drive (left and right) variables, the blue arrow shows the USonic device measuring distance to obstacles, the green arrow is a flag to indicate an object has been detected.



The Tasks work as follows.

- **Wander.** A series of movements. This task sets the motor speeds and the time for all movements. It outputs the drive speeds.
- **Evade.** This task listens to the USonic ping and if an obstacle is close it output a series of movements, again setting motor speeds and also duration. It also sends a flag to the arbitrator to indicate an obstacle has been detected
- **Arbitrate.** This selects the motor speeds either from Wander or Evade based on the state of the flag sent by the Evade task. It then sends them to the Drive Motors task
- **Drive Motors.** This sends the motor speeds it has received to the servos, to physically turn them.

First, we'll have a look at a couple of sketches to prepare; the first is about driving motors and the second looks at the Ultrasonic sensor

1 Preparation (1) Driving the motors

(a) Open up the sketch **RTOS_MotorTest_Rel**. You will see the following tasks already coded:

- **MotorTest.** This assigns values to **driveL** and **driveR** (in shared memory – global variables) which are then read by
- **MotorDrive**, this task sends driveL and driveR to the servos
- **Monitor.** This task reads the values of driveL and driveR and sends them to the serial monitor.

Run the sketch and see what happens

(b) Change the code in **MotorTest** to obtain your favourite robot behaviour.

(c) Check out the details of the code. Look for task priorities and check that each task can be put into its *blocking* state.

2 Preparation (2) The Ultrasonic sensor

(a) Plug the USonic sensor into the proto-board so it is facing forward. Connect Vcc with a red wire to 5V, connect Gnd to Gnd with a black wire, connect Trig to pin 9 with a yellow wire, and connect Echo to pin 2 with a green wire.

(b) Open up the sketch **RTOS_USonic_Rel.** Here there is one Task and also a software Timer. Run the sketch, open up the serial monitor and make sure it works.

(c) To understand the code, look for the following

(i) Code which creates a software timer. This timer calls the **timerCallBack()** function which calls a function to emit an USonic ping. The function is craftily hidden in a #define.

(ii) Attaching a hardware interrupt to pin 2 to receive the echo signal from the USonic sensor. Look for the Interrupt Service Routine connected to the interrupt. Check out how it responds to the rising edge of the echo signal and then the falling edge. See how it calculates the pulse length in microseconds. This is the time between the ping being emitted and received and is used to calculate the distance **dist** to the obstacle.

(iii) Look where the variable **dist** is being used.

(iv) What does the **Monitor** task do?

3 Coding the Subsumption Architecture

Much code is provided in the template sketch **RTOS_Subsumption_Rel.** All you need to do is follow the numbered instructions in that sketch.

You can cannibalize the previous two sketches. Just holler if you get stuck.