Faculty of Science and Engineering

Referred Coursework – 2019/2020 Academic Year

PLEASE NOTE: If you have been referred in the COURSEWORK element of this module and are required to be reassessed by COURSEWORK, please complete this referred work.

Module Code:	ELEC350
Module Title:	Advanced Embedded Programming
Module Leader:	Nicholas Outram
School:	Engineering, Computing and Mathematics

DEADLINE FOR SUBMISSION: 3pm on Thursday 20th August 2020

SUBMISSION INSTRUCTIONS FOR CANDIDATES
Referred coursework must be submitted electronically using the online submission facility in the DLE by the published deadline. If you have any queries on submission or in relation to the referred work, please contact the Module Leader in the first instance, if they are unavailable please contact the Faculty Office on 01752 584584 immediately so any problems can be rectified.

If you require any part of this publication in larger print, or an alternative format, please contact:

Faculty of Science and Engineering
T: +44 (0) 1752 584584
F: +44 (0) 1752 584540
E: science.engineering@plymouth.ac.uk

Software Required

Due to the Covid-19 situation, this referred coursework has been rewritten to remove the requirement to use any hardware.

You will need the following:

Microsoft PowerPoint (part of your Microsoft365 package)

Video demonstration tool

- If you choose to provide a pre-recorded video, screen recording software. You can use Microsoft Stream which is a web app provided with Microsoft365 or another package.
- If you prefer to give a live presentation, you must arrange to meet online with the tutor using Microsoft Teams or Zoom

If you need support with setting these up, please contact the module leader as soon as possible.

Multi-Tasking

A central theme in ELEC350 and ELEC351 (Advanced Embedded Programming) is multi-threaded programming. For this we used a real time operating system which is part of Mbed-os.

For this assignment, you are to record or deliver threes presentations, one on each of the topic's outlined below. For each topic, your task is to give a presentation as if you were teaching it to new students who have just enrolled on the module.

You should produce *at least* one power point slide per sub-topic (or more) and record a presentation with audio narration. You can do this offline, or online using Zoom or Teams.

Topic 1 – Multithreading

This is worth 25% of the overall mark

Give a presentation on multi-threading, including the following subtopics:

- What it is
- Why it is needed and what problems it solves
- Benefits for the programmer
- Overheads and disadvantages
- The concept Round-robin scheduling
- The basics of how a round-robin scheduler works

Do not simply quote text from another source. You should use your own words and quotes from others "sparingly". Remember you are explaining this to beginners.

Topic 2 – Multi-threaded primitives

This is worth 25% of the overall mark

Give a presentation on the following key multi-threaded concepts and primitives.

- Mutex Locks
- Counting Semaphores
- Signal-wait (now known as event flags)
- Condition variables

For each, explain their purpose, where/when they can be used and give an example of how it is used.

Note - as this is a level 6 module, it is expected you can work with state of the art. Condition variables are new in Mbed-os, but not a new concept. You should research this yourself.

Topic 3 – Case Study

This is worth 50% of the overall mark

Figure 1 depicts a microcontroller unit (MCU) based real-time system designed to monitor driver behavior in a car

Driving Events – The system is intended to detect the following energy inefficient and dangerous driving events:

- Low speed, rapid press on the accelerator pedal
- High speed, rapid press on the brake pedal
- Very high acceleration events (impact detection) using an accelerometer
- High steering angle at medium or high speeds

Input Sensors

There are a number of input sensors on the device

- Speed is measured from a sensor on the wheel shaft and produces a short pulse every time one of the wheels rotates 360 degrees. This is interfaced to a GPIO pin on the MCU.
- The accelerator and brake pedals are mechanically connected to potentiometers which produce a voltage proportional to how far the pedal is depressed. These are interfaced to the MCU using separate ADC inputs
- The steering angle is also connected to a potentiometer and is interfaced to the MCU using an ADC input.
- There is an on-board accelerometer connected to the MCU via the SPI interface

Output Devices

There are also three output devices on the device:

Buzzer – this is sounded to alert the driver of a negative driving event. It is interfaced using a simple GPIO line.

Ethernet – All driving events are sent to a listening TCP/IP socket on central computer in the car. This enables the driving events to be logged and warnings displayed on the electronic dashboard.

SD Card – Any system faults detected are logged to an SD card that is connected via an SPI interface. This is used by garage engineers to check for faults and in a future release, to upgrade firmware.

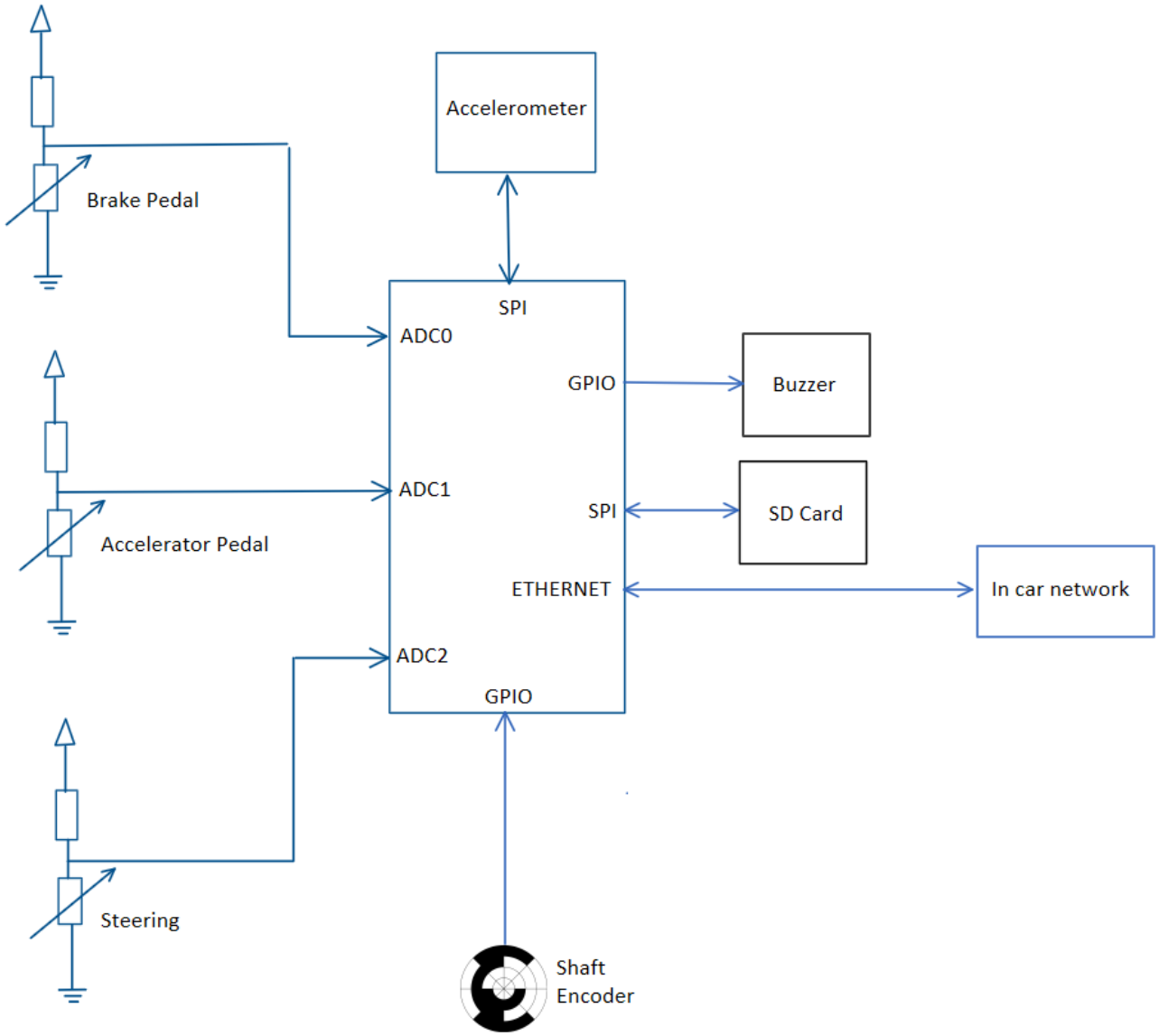

Figure 1. Car Driving Behavior Monitor

Your task is to consider and present how each device should be monitored. In your presentation you should consider the following:

- What real time software techniques you might use. Consider whether you would use polling methods, a thread or interrupt (or some combination), and if so, how the priorities would be

decided and managed. If sampling, do you need a fixed sampling rate and how would you achieve that? Does data need buffering in ram?

- How you can ensure events are not missed
- How to ensure the system is responsive
- Which Mbed-os classes you might use

You do NOT have to produce any software, but you may if it helps to illustrate.

Submission

This is individual work.

Your slides and any additional files must be submitted online as a single ZIP archive by the deadline shown on the DLE.

This must include your PowerPoint slides and any links to video recordings

Please be aware of the university regulations on plagiarism.