Embedded System Design

Assignment 2

Name – Ayush Agarwal

Roll No – 20095021

Branch – Electronics

Ayush Agarwal

Section A

Level 1: Theoretical Description

The first step of Embedded system design involves interacting with the clients and key stakeholders to understand the problem statements and the design requirements , since all solutions are engineered keeping the customer needs in mind . The designer discusses with the end-user , and identifies the requirements of the embedded system , the constraint factors such as **PPA metrics** of the embedded chip (Power , Performance , Area) , the cost that the client can pay for such a system , and manufacturability . The designer also comes up with a list of functional requirements for the system , which must be fulfilled .

Level 2: Pen-and-paper based Creative-Design

Based on the required feature set and the customer requirements , the designer makes a pen and paper based mechatronic model of the required embedded system . This involves thinking about the conceptual architecture and arrangement (as well as wiring strategies) of different categories of blocks , as well as thinking and breaking the different system feature requirements into rough blocks (which would be made more precise in the next step) . The different categories would be to **break the system into sensors , actuators , communication , processing , signal improvement elements as well as breaking the computational problem into a rough digital , analog and power class of problems** . This would require an expert class of embedded electronics system designer .

Level 3: Block-level Description Diagram

Once the conceptual design is complete, the designer should create a block-level diagram that shows the different functional blocks of the system and how they are connected. This diagram should provide a high-level view of the system, showing the different modules and how they interact. At this stage, the designer should also identify any potential problems or bottlenecks that may affect the overall performance of the system. The rough explanations given in the previous steps are now broken into systems , which are further broken down into subsystems until the point we can extrapolate them to a set of solved problems . Then we would have defined the wiring arrangement of each of the blocks precisely , as well as we would only have to think about the parameters of each block along with its effect on the whole system .

Level 4: Detailed Circuit-level Design

Once the block-level diagram is complete, the designer must start working on the detailed circuit-level design of each module. Each block is now replaced by a circuit along with the parameter values required for it . The parameters can be as simple as values of R , L or C , or something bigger like the feedback levels , heating etc . This involves selecting the

Ayush Agarwal

appropriate components and designing the circuits to meet the performance requirements. The designer must take into account factors such as power consumption, noise, and signal integrity, and ensure that the circuits are designed to be reliable and robust.

Level 5: Layout-level Design

After the circuit-level design is complete, the designer must create the layout-level design of the PCBs or System-On-Chip (SOC) modules. This involves creating the board file of the PCB , which would include steps such as **Power Planning , Pin Planning , Floor Planning , Clock Tree Synthesis , Static and Hold timing analysis , Routing via Steiner tree synthesis , parasitic factor modelling and accounting , optimisation etc .**

This involves creating a physical layout of the components on the board, taking into account factors such as the size of the board, the location of the components, and the routing of the traces. The designer must ensure that the layout is optimized for manufacturability, reliability, and performance.

Level 6: Product-level Design, Manufacturing and Final-Testing

The final stage of the electronics hardware design process involves product-level design, manufacturing, and final testing. At this stage, the designer must ensure that the design is optimized for mass production and that it meets all relevant industry standards and regulations. The designer must also work closely with the manufacturing team to ensure that the design can be manufactured efficiently and reliably. Finally, the designer must oversee the final testing of the product to ensure that it meets all performance and quality standards.

In conclusion, electronics hardware design is a complex and multi-faceted process that requires a broad range of technical skills and knowledge. The six levels of design-creativity provide a framework for understanding the different stages of the design process, from initial problem definition to final product testing. By following this framework and working closely with other members of the design team, an electronics hardware designer can create robust and reliable designs that meet the needs of their customers and end-users.

Ayush Agarwal

Section B

1. Electronic Unit to control the Motors in a DRONE

Input signals will be signals sent by user via remote control to move the drone – so we will need a **communication module** with low network latency for smooth control of the drone .

Since the drone is a UAV and is going to be battery connected , our circuit would need to be optimised for low energy application too .

Since the signals sent by the user are going to be basic RPY , hence we would need a small microprocessor or **logic unit** to **convert** the incoming Roll-Pitch-Yaw **(RPY) signals to** the values suitable for **motor inputs** .

We would also need **optical encoders** for getting the exact speeds of the drone motors (although we could also get that using the **back-emf** , it would be slightly more difficult) . These would be essential for feedback to the controller , so as to allow advanced features like disturbance avoidance .

We would also be needing **IMU sensors** which would have accelerometers for measuring speeds and gyrometers for making the **precise odometric model** of the drone , and thus achieve **better kinematic control** . These modules could also help in calculation of the take-off voltage required for a certain payload , which could be useful for a **variable payload situation drone** such as a delivery agent drone . This could also be used for introducing **fault tolerance** in the drone , as a deviation in the proportionality between the applied voltage and the current requirement feedback could be a major indicator for a damaged propellor , since the load torque for that propellor would change . In such cases , an **emergency landing protocol** could be initiated .

For **disturbance avoidance** of the drone from external factors such as wind and thus to balance the drone , we would also need a **controller** in the loop , which could be as simple as **PID** (Proportional-Integral-Derivative) or as advanced as a **MPC** (Model-Predictive-Controller) . The controller can be implemented digitally also inside the main microprocessor module using **digital controls theory** .

Talking about the motor itself , it could draw huge amounts of current thus separate pins are needed in the power supply module , and **separate module** is needed in the main microprocessor to avoid burning up the flight controller altogether . Also , we would need **heat-sink** structures with decent exposure to air to avoid burning up the drone mid-air . From a mechanical perspective , we would also need **sufficient ventilation** on the motors too . Also , we would need **H-bridge structures** (for directional change thus improving controllability) along with **buck-boost amplifiers** (for amplifying the voltage from onboard power supply for motor usage purposes and for low latency variable voltage signal generation) for the powertronic design of the motor driver module

For **collision avoidance** purposes of the drone , we could require **radar or lidar** like structures , so that the motors can be controlled in a way to protect the drone from collision from say a bird or another drone . This could be coupled with a much simpler warning system to the user , or an automated path planning approach for which suitable algorithms can be devised accordingly .

2. Digital Weighing Machine used in hospitals/clinics to measure weight of patients

The sensor used for measuring weights is called as a **load cell** . We would be required to choose a load cell among huge number of possible cells available in the market , based on the **temperature range** of operation of device (taking a reasonable range , and choosing extreme case temperatures only if we are designing a system for usage in extreme climate conditions such as polar region cold or desert like hot) , and also based on **the expected weight range** (which we can reasonably assume to be around 0-140kgs , with higher probabilities between 55-95kg range) .

Since **Load Cells with higher capacities would be more costly** , one intuitive approach could be to use **multiple load cells** in the structure and distribute them **symmetrically** (so that their individual readings are also in a low range of capacity) . This would lead to lower load per load cell, which would mean we could use lesser capacity load cells and thus reduce the costs . However do note that the errors in the load cells might add up also , hence we cant use a huge number of load cells otherwise the variance of our readings would increase if the variance of a single load cell wasn't small enough . Later , we can add the values in all the load cells .

Next , we would need **ADC** for analog to digital conversions , for feeding the sensors readings to the central microprocessor . The **number of bits** we convert the data to completely depends upon the reading ranges , as well as the final value ranges (which would reasonably be a **3 digit number with 2 decimal places**)

Next we would also need to think of the **LED or the LCD screen** , or if the device has to be made **cheap**ly , we can use **7 seg-displays** also . We would also need **driver circuits** for these , since they might or might not draw a good amount of current from the power source .

We would also need to make an **ON/OFF button** , so that the device can be switched off and kept into storage without dissipating the energy stored in the cells (assuming that the digital weighing machine is not wired but a battery operated device since I had such a model at my home . This button would work with the power module of the circuit .

We would also need to think of the **"SET ZERO" button** to remove zero errors , if any . This would help in accounting for atmospheric pressure variations and any dust weight on the machine too . The signal would go to the central microprocessor , which on the inside using **software programming** set the present output of load cell readings as zero .

3. Infotainment System in Cars

For such a device , **user experience UX** is pretty essential component . Sure we would need to make a **touchscreen** , along with the **screen driver** , but we would also need to make sure that what's displayed on the touchscreen is also arranged in a nice and colourful way .

We would need to have **multiple ports** for connection .

We would need **AUX cable** for fast transfer of music to the central microprocessor which would in turn play it .

We would also need **power ports** , along with multiple **charging wire** types(multiple as we usually have more that 1 person at a time requiring phone charging on any long road trip nowadays) (preferably **USB-C** as the world is moving towards it) so that people in the car can charge their

Ayush Agarwal

phones using the central dashboard which we call the infotainment system. Care should be taken to make wider wires for higher current drawing capacity , along with **MCB**(Miniature Circuit Breakers) so as to provide **Overload Protection** since careless people might just connect things other than phones .

The infotainment system would also need a **comms module** , so as to allow **wireless integration** (as well as **automatic detection**) of the driver's phone . **Good software design** would be needed so as to allow smooth integrability between the phone and the embedded system , and so that all the calls , messages can be smoothly transferred to the infotainment system to be handled without having privacy issues and whilst controlling the songs also . The communication could be preferably **Bluetooth** , since Bluetooth works well within low range of operation .

Another idea I had uniquely is that we can make this embedded system as a wifi router too. Since the Car has bigger space than a phone , hence we could fit bigger antennas , hence catching even fainter and lower energy signals , thus being a good **wifi catcher and supplier**.

The system could be connected to **temperature sensors** across the car to display the internal and external temperatures on the screen . It could also be connected to light sensors , thus allowing night mode on screen whenever there is night .

The system could also have a **GPS chip** for allowing it to behave as a **navigator** , and have a good software stack so that users can use it .

Finally , but coming to the most important thing , we need to connect this chip to a speaker , with wider wires so that larger currents could be drawn . We would also need **DSP module** in the processor , so that Bass , Treble , Pitch and Equalization could be set accordingly by advanced users who need to fine tune the sound and remove the buzz .

We would also need **AM and FM receivers** and signal processing elements so that the users can listen to songs which are being broadcasted freely .

Adding an on screen AC controller with advanced selections is a possibility , but the hardware implementation would be better for it , and it may or may not be made a part of the infotainment system and instead be solved on a different chip altogether .

Finally we would need to check whether the engine is on or off using different sensor , and switch the **power source** between the **alternator and the car battery** . We would also need emergency message display if the car battery is running low .

Ayush Agarwal

<u>4. Digital Camera</u>

I would divide this question into designing of the lens and designing of the rest of the camera , since much focus needs to be given to the first part .

The **lens** basic structure is made up of **3 lenses** – 2 concave and 1 convex . The distance between the 2 concave stays fixed . We need to make 2 electromechanical actuation systems for moving the lenses , **one** for moving the convex lens relative to the other two for **zooming** in and out of the image , and the **second** system for moving this set of 3 for changing the **focus** of the image , so that the focus lies on the sensor plane . However this system would often be controlled by the Auto Focus (AF) module rather than the user themselves . Care has to be taken so that the thickness of these lenses add up in such a may so as to avoid **chromatic abbertion** and the colours don't split in the image .

We would choose a **mirrorless system** over a **Digital Single Lens Reflex (DSLR)** so as to reduce the size of the overall device . We also need to make the **Anti-Shake mechanism** within the lenses to increase the image stability and avoid blurring due to unsteady motion of the hand .

Next , we need to design the **electromechanical system** for changing the **aperture** of the lenses , which would be controllable by the user . We also need to add a **shutter** and make another mechanism for controlling it . I have not discussed the exact details of such mechanisms , simply to avoid the answer becoming too long , however , it is of significant concern to an embedded system designer .

We also need to make the image splitters so as to split the rays between the electronic **Optical View Finder (OVF) , the Auto Focus (AF) module and the CCD/CMOS array** . The OVF needs to be connected to a small screen (more like window) whereas the rest of the ports indirectly connect to the hardware .

Now coming to the rest of the camera , most cameras use a **CCD (Charged Coupled Device)** or newer ones use **CMOS** based **image sensors** . These would be connected to **ADC** , which would send this digitized image to a processor . There , we would have a **signal processing circuit** , where we can vary the contrast , saturation , highlights , brightness , exposure and shadows using user inputs of how much they want to apply it , intelligent ML based filters which do the optimum hyperparameter search for auto-enhancements , using algorithms coded in the hardware and processed by specially modified parallel computing based solutions in the hardware .

Finally , the output of the processor would be connected to the **SSD** (Solid state Drive) **memory** which can store these images and videos .

There would also be interfaces in forms of **buttons** for some of these features (like there would obviously be a button for clicking images , one for flash etc) . The exact number of buttons would depend upon how much convenience designer wants to deliver vs how much space is available on the device . The screen would have to be **touchscreen** so that we can accommodate rest of the controls upon it .

When we are viewing the images , there has to be a **cache memory** which pre loads the nearby files too , so as to enhance the delivery of images to screen and reduce latency .

There would also be a **strong light source** on the device which needs to be interfaced in this embedded system , for the purpose of flash . Its **driver circuit** would have tunable monoshot multivibrators , capacitors etc .

Also , there would be a **power button** on the camera , and we need to design the power module too . Care has to be taken to make this device power optimised since it is going to be a battery operated camera .

Avush Agarwal