EE23DP001
Aditya Shirodkar

EE690: Embedded Systems Design
Homework 2

Q) Develop the architecture for an embedded system to control a windshield wiper system. A windshield wiper motor assembly is available and has one digital in (RUN) and one digital out (HOME). As long as the RUN line is held high, the motor runs and the wiper operates continuously, i.e wipes back and forth. Whenever the wiper is in the lowest position, the HOME line goes low, when the wiper is elsewhere it is high.

Your task is to design an embedded controller that is capable of operating this wiper in three modes -

a) continuously wiping (for heavy rain)
b) intermittent (wipe once, wait, wipe once, wait, etc)
c) wipe exactly once and then stop

Your design should also include how you will take input from the user (the driver of the vehicle) in an intuitive way- i.e. the user should not need to know the internal workings of the system. You should also ensure that the wiper is never stopped in the "up" state- i.e. your controller should always make sure to run the motor till the wiper comes down and then stop.

A)

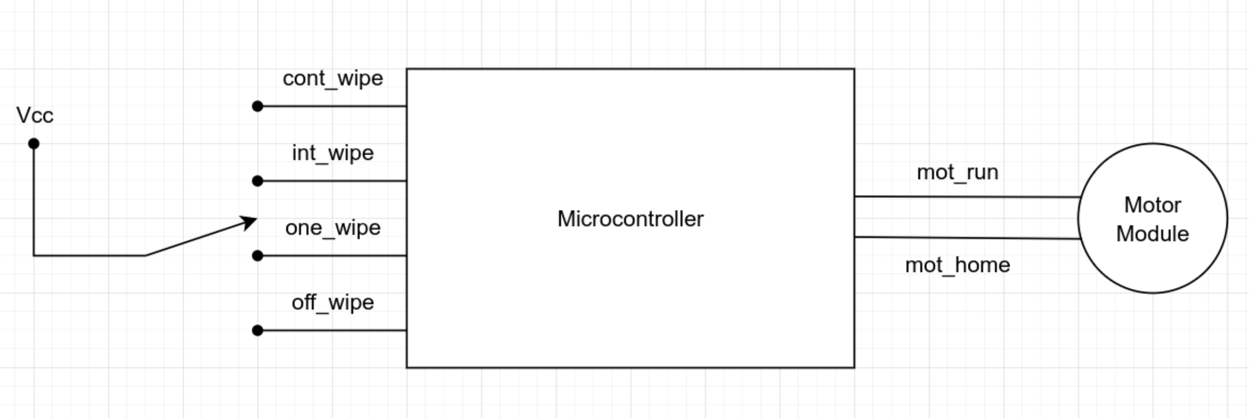
Inputs to user will be provided using a wiper control stick, with three options:

1. Rain: Triggers continuous wipe
2. Drizzle: Triggers intermittent wipe
3. Clean Window: Triggers single wipe

- Internally, each option is selected through an electrical contact switch, which then feeds a signal to the microcontroller.

- Let the naming of pins be as follows:
    - Continuous wipe: cont_wipe
    - Intermittent wipe: int_wipe
    - Single_wipe: one_wipe
    - Motor RUN: mot_run
    - Home: mot_home

- Schematic:



- PseudoCode:

```
volatile int state = 0;
bool cont_wipe, int_wipe, one_wipe, mot_home;     //Inputs
bool mot_run = 0;   //Output

void state_assign_interrupt(void)  //Same as GPIOHandler
{
/* state_assign_interrupt is written such that upon receiving an input from user, and interrupt is triggered. States are assigned such that state = 0 is off, state = 1 is cont_wipe, state = 2 is int_wipe, state = 3 is one_wipe
*/
}




void main()
{
```

```
/*Config GPIO to have cont_wipe, int_wipe, one_wipe,
mot_home as inputs and mot_run as output */
gpio_config();

while(1)
{

if(state==0)        //Off
{
     if(home==0)
     {mot_run=0;}
     else
     {mot_run=1;}
}

else if(state==1)   //Continuous Wipe
{mot_run=1;}

else if(state == 2) //Intermittent Wipe
{
     while(state==2)
     {
     mot_run=1;
     if(home==0)
     {mot_run=0;}
     else
     {mot_run=1;}
     delay();
     }
}

else if(state == 3) //Single Wipe
{
     mot_run=1;
     state=0;       //Turn off
}

}
```