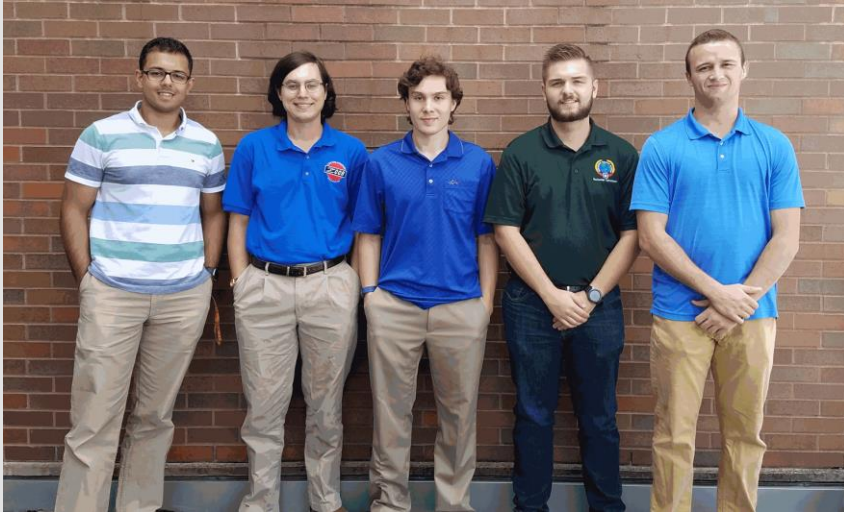


# MSD Team 20225: The Doc B Racing Real-Time Telemetry System

## The Telemetry Bois

Adam Bork, Sahil Gogna, Josh Henderson,  
Michael Schroeder, Nick Washco

## Introductions:



Member	Role	Contact
Nick Washco	EE Lead	<a href="mailto:njw6366@rit.edu">njw6366@rit.edu</a>
Josh Henderson	Facilitator	<a href="mailto:jh2290@rit.edu">jh2290@rit.edu</a>
Sahil Gogna	Project Manager	<a href="mailto:sxg4060@rit.edu">sxg4060@rit.edu</a>
Michael Schroeder	CE Lead	<a href="mailto:mjs5127@g.rit.edu">mjs5127@g.rit.edu</a>
Adam Bork	Team Liason	<a href="mailto:acb6075@rit.edu">acb6075@rit.edu</a>

## Agenda:

- Team Vision for Build and Test Prep Phase
- SD Write
- Queue Management
- Board Changes
- Test Plans
- Risk Assessments Review
- Plans for the Next Phase
- Q&A

## Team Vision for Build and Test Prep Phase

- **Make Test Preparation documentation for what needs to be tested, and how it should be done.**
- **Focus on SD Card Read and Write Functionality.**
- **Start testing and see if any risk change.**
- **Continue to work on enclosure redesign.**
- **Research WIFI and Bluetooth capability with the PCB and microcontroller.**

# Testing- SDWrite() from MSD I

```
void SDWrite(uint8_t * buffer)
{
    FIL MyFile;
    FRESULT res;
    uint32_t byteswritten;

    sprintf(fileNameVar, "run%02d.csv", count4);

    if(HAL_GPIO_ReadPin(ENABLE_SW_GPIO_Port, ENABLE_SW_Pin) == 1){
        if (off_to_on_pulse == 0)
        {
            ResetQueues();
            off_to_on_pulse = 1;
        }
        on_to_off_pulse = 0;
        HAL_GPIO_WritePin(LED1_GPIO_Port, LED1_Pin, GPIO_PIN_SET);
        res = f_open(&MyFile, fileNameVar, FA_OPEN_APPEND | FA_WRITE | FA_READ);
        if (res != FR_OK)
            Error_Handler();
        res = f_write(&MyFile, buffer, strlen(buffer), (void *)&byteswritten);
        if ((byteswritten == 0) || (res != FR_OK))
            Error_Handler();
        f_close(&MyFile);
    }
    else
    {
        if (on_to_off_pulse == 0)
        {
            count4++;
            HAL_GPIO_WritePin(LED1_GPIO_Port, LED1_Pin, GPIO_PIN_RESET);
            on_to_off_pulse = 1;
        }
        off_to_on_pulse = 0;
    }
}
```

```
void SDWrite(uint8_t * buffer)
{
    FILE_F* MyFile;

    sprintf(fileNameVar, "run%02d.csv", count4);

    if(HAL_GPIO_ReadPin(ENABLE_SW_GPIO_Port, ENABLE_SW_Pin) == 1){
        if (off_to_on_pulse == 0)
        {
            ResetQueues();
            off_to_on_pulse = 1;
        }

        on_to_off_pulse = 0;
        HAL_GPIO_WritePin(LED1_GPIO_Port, LED1_Pin, GPIO_PIN_SET);
        if(!f_open(MyFile, fileNameVar, FA_OPEN_APPEND | FA_WRITE | FA_READ))
            Error_Handler();
        if(f_write(buffer, sizeof(buffer[0]), strlen(buffer), MyFile) != strlen(buffer))
            Error_Handler();
        f_close(&MyFile);
    }
    else
    {
        if (on_to_off_pulse == 0)
        {
            count4++;
            HAL_GPIO_WritePin(LED1_GPIO_Port, LED1_Pin, GPIO_PIN_RESET);
            on_to_off_pulse = 1;
        }
        off_to_on_pulse = 0;
    }
}
```

# Testing: Queue Management

```
void WriteQueues(uint8_t* buffer1, uint8_t* buffer2, uint8_t* sensorBuff)
{
    int count = 0;
    switch (buf_choice)
    {
        case 1:

            for (count = 0; count < SD_CARD_BUFFER_SIZE; count++)
            {
                // Copy byte-by-byte
                memcpy(buffer1[count], sensorBuff[count], sizeof(uint8_t));
            }

            // DMA Write to SD Card
            SDWrite(buffer1);
            buf_choice = 2;

            break;

        case 2:

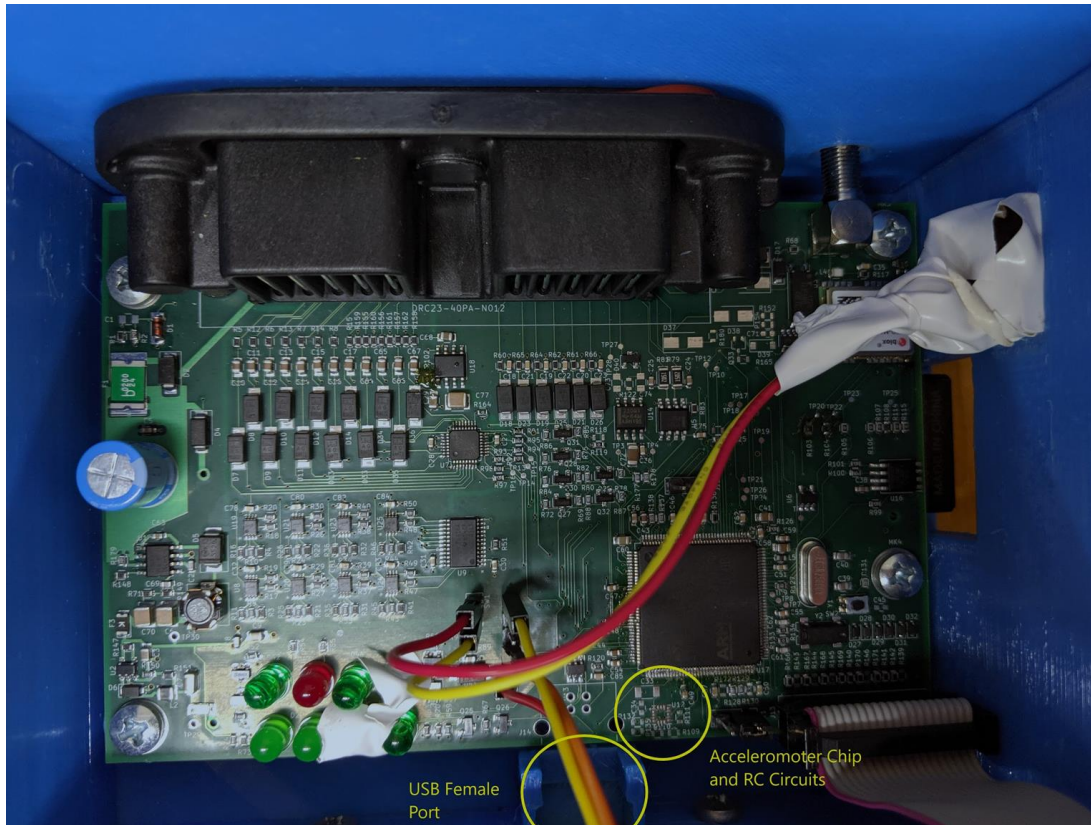
            for (count = 0; count < SD_CARD_BUFFER_SIZE; count++)
            {
                // Copy byte-by-byte
                memcpy(buffer2[count], sensorBuff[count], sizeof(uint8_t));
            }

            // DMA Write to SD Card
            SDWrite(buffer2);
            buf_choice = 1;

            break;

        default:
            break;
    }
}
```

## Board Changes with SMT Lab



- WiFi and Bluetooth not compatible with current PCB design
- Would have to redesign PCB to add low-power wifi module to the board.
- Chosen not to redesign, and focus on the Sd card functionality instead.
- Received all parts ordered during MSD I except a jumper resistor.
- Once resistor comes in will work with SMT Lab to solder parts to the PCB and test the accelerometer.

# Test Plans

- **Purpose:**
  - The purpose of the testing plan this phase was to implement and test viable SD Card Read/Write methods as well as debug logical analyzer ports on the microcontroller.
- **Instructions:**
  - A meeting with Dr. Carlos Barrios from the Electrical Engineering Department assisted us in coming up with more ways of implementing the SD Card Read/Write method via analyzing logical analyzers on the board.
  - Hardware schematics were analyzed to discover where the logical analyzers were
  - Understand more of the code-base implementation



# Test Plans Documentation

P20225 Test Plan				
IO Tests				
Name	Description	Purpose	Target Completion	Personnel
SD Test	Save text file to SD Card	Test SD save function	2/13	Mike, Sahil
Serial Test	Communicate with PC via terminal	Test USB functionality	2/13	Mike, Sahil
Config File	Create Config and pass to board	Test system adaptability	2/13	Mike, Sahil
Hardware Tests				
Name	Description	Purpose	Target Completion	Personnel
Accelerometer	Pass voltages to accel to trace output	Test function and validity of accelerometer	2/13	Josh, Nick, Adam



# Risk Assessment (Remained the same)

Risk Type	Risk Item	Effect	Risk Assessment			Action to Minimize Risk
			Likelihood	Severity	Importance	
Electrical	Short board in testing/use	Board is unusable	2	3	6	Practice safe testing, be careful of exposure
Electrical	Sensor damaged/ does not work	Data from sensor not able to be read	1	2	2	Ensure sensors are set up correctly and test functionality
Electrical	Sensor not calibrated correctly	Invalid/incorrect data	2	1	2	Verify calibration procedure is closely followed
Electrical	Wiring Harness/Connections fail	Sensor data lost	1	2	2	Practice careful wiring procedures
Electrical	Electric Shock	Harm to team member, component damage	1	2	2	Install properly and verify team members take precaution
Electrical	Adding new Accelerometer and USB port	Possibly board damage, further rework	1	2	2	Verify understand circuit before installing, work with SMT lab personel
Electrical	Digital and Analog Sensors don't integrate to embedded system	Full data set not able to be recorded	1	2	2	Test all sensors on system
Mechanical	Enclosure not Secure	Potential Damage to Enclosure/Car	1	3	3	Follow instalation instruction
Mechanical	Enclosure too close to Engine Bay	Components on board damaged due to heat	1	2	2	Installation of enclosure in car far enough away from engine bay
Mechanical	3D print not finished completley	Time and Monetary loss, not able to implement new enclosure design	2	2	4	Verify 3D model sent is correctly formatted
Mechanical	Enclosure exceeds temperature value designed to withstand	Sensor Failure/Board damage	2	3	6	Add heat wrap to ensure enclosure is able to withstand most extreme temperatures
Planning	Not enough time to fix code base and to save data	Product not complete	1	2	2	Work on schedule, don't fall behind

Planning	No access to track to test device	Unsure of how system behaves in a real test environment	2	2	4	Look for oppurtunities to test, test product in own cars prior to at the track
Planning	Not having Demo Ready for Imagine RIT	Not able to meet requirement for MSD 2	1	3	3	Stay on task and work according to schedule
Software	Bugs in code	Code doesn't perform as expected	2	2	4	Verify functionality in simulation
Software	SD Card functionality not working	SD Card data not able to be saved/read	2	3	6	Finish code and work out bugs/ test functionality
Software	Embedded data not compatible	incorrect data being analyzed	2	1	2	Verify proper integrations between system componets
Software	Not able to fix bottlenecking in code	Unable to read/write data	2	3	6	Work with Subject Matter experts, ensure testing time/environment is available for code

## Plans for Next Phase: Subsystem Build and Test Phase

- Continue testing and functionality of SD card.
- Start writing new code for SD card, stop using previous teams code.
- Add new parts to the PCB, and test the accelerometer.
- Update designs, flowcharts, and risk assessments.
- Continue to work on the Gantt chart.
- Continue with redesign of enclosure.
- Test system in car if SD testing is completed.

**Thank You from  
The Telemetry Bois**

**Questions?**