



# Doc B Racing Telemetry Acquisition System



Telemetry Bois

---

# Background and Motivation



---

# Motivation

- The key to going faster is determining through quantitative data the best setup for your car
- Many currently available telemetry systems are too expensive for the average racer
- Vintage race cars are often incapable of making use of these systems
- Manufacturer specific sensors cause limitations in the data that can be collected

---

# System Goals



---

## Expected Outcomes

- Support wide array of new and old sensor technology
- Transfer data via SD Card
- System able to handle environment successfully
- Design a cost competitive system to compete at the retail level (<\$1000)



# Hardware



# Embedded Telemetry Device

Custom embedded system uses the following:

- STM32F407ZG Processor
  - Cortex M4 at 168MHz
  - 2 UART, 2 I2C, 2 SPI, SDIO, CAN, USB FS Device
- 2MB SRAM
- 16MB SPI Flash
- ADS124S08 External 12 input, 24-bit ADC
- ublox NEO-M8N GPS





**Software**





# Queue Management

- Inverted Buffer Scheme

```
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;
    }
}
```



# SD Read/Write

- Basic Read/Write accomplished between board and SD card

```
void SDFileTest(void)
{
    FIL MyFile;
    FRESULT res;
    /* FatFs function common result code */
    uint32_t byteswritten, bytesread; /* File write/read counts */
    uint8_t wtext[] = "Hello, STM32F4 FatFs! Hey it works!!!"; /* File write buffer */
    uint8_t rtext[100]; /* File read buffer */
    //printf('SD File create\n');
    // Create file test
    if (f_open(&MyFile, "STMTXT", FA_CREATE_ALWAYS | FA_WRITE) != FR_OK)
        Error_Handler();
    res = f_write(&MyFile, wtext, sizeof(wtext), (void *)&byteswritten);
    if ((byteswritten == 0) || (res != FR_OK))
        Error_Handler(); f_close(&MyFile);
    //printf('done!\n');
    //printf('SD File open\n');
    // Read file test
    if (f_open(&MyFile, "STMTXT", FA_READ) != FR_OK)
        Error_Handler();
    res = f_read(&MyFile, rtext, sizeof(rtext), (UINT*)&bytesread);
    if ((bytesread == 0) || (res != FR_OK))
        Error_Handler();
    f_close(&MyFile);
    if ((bytesread != byteswritten))
        Error_Handler();
    //printf('done!\n');
}
```



# Final Outcome



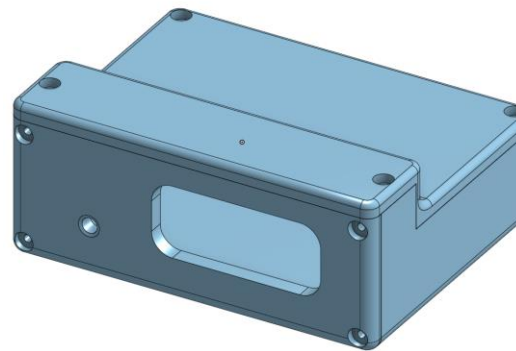
# Final Outcome

## Sensor Support:

- 12 analog inputs (Temperature, Pressure)
- 6 digital inputs (Speed, Rotation, Switch)
- CAN bus (Tire Temp, Accelerometer, etc)
- GPS
- Tachometer input (direct reading from coil)
- Internal temperature and input voltage

## Features:

- Writes data to SD card
- Reconfigurable via configuration file on the SD card
- Well designed enclosure
- Proper SD Card Queue Rate 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;
    }
}
```

---

# Acknowledgements



---

# Special Thanks

- Our Main Sponsor: Dr. Thomas Bork
- Our MSD Guide: Harold Paschal
- Dr. Carlos Barrios
- All our Technical Sponsors
- The Team:
  - Adam Bork
  - Josh Henderson
  - Michael Schroeder
  - Nick Washco
  - Sahil Gogna