# NFPA Fluid Power VEHICLE Challenge

NFPA Education and Technology Foundation

**FINAL PRESENTATION**
**FLUID POWER CLUB AT**
**SOUTH DAKOTA STATE**
**DOUGLAS PRAIRIE**
**4/29/2025**

# Team Introductions:

Jake Druley

Jacob Hinders

Gabe Nelson

Carson Gunnerson

# Design Objectives

- New Frame
  - Lighter material than thick steel frame
- New Components
  - Pump, Motor, Manifold
- Simplify hydraulic circuit
- Improving effectiveness of 6-Speed Transmission
- Fixed position components to the frame

# Progress Since Midway Review

- New Frame Construction
  - Chromoly steel frame
  - Routed Plumbing
  - Attached Components
- Test New Bike: 4/14/24-4/25/24
  - Collect Data
  - Analyze Data
  - Make Improvements
- Finishing Touches: 4/26/24-4/29/24

# Design Choices

- Improved manifold; size and circuit
  - Simplify tubing
  - Reverse mode
  - Smoother pressure dump
- Transmission
  - Use full range of the Pinion 6-speed transmission
- Pump / Motor
  - Changed to a smaller gear pump
  - Decreased motor displacement to 2.8 cc/rev
- Lowered seat position for improved leverage on pedals

# Back Mount Design

- Removed last years "pallet" design
  - Improved ease of access (simplified to just nuts and bolts)
  - Saved space and weight
  - Simplicity
  - Safety (last year the pallet design rattled everything loose)

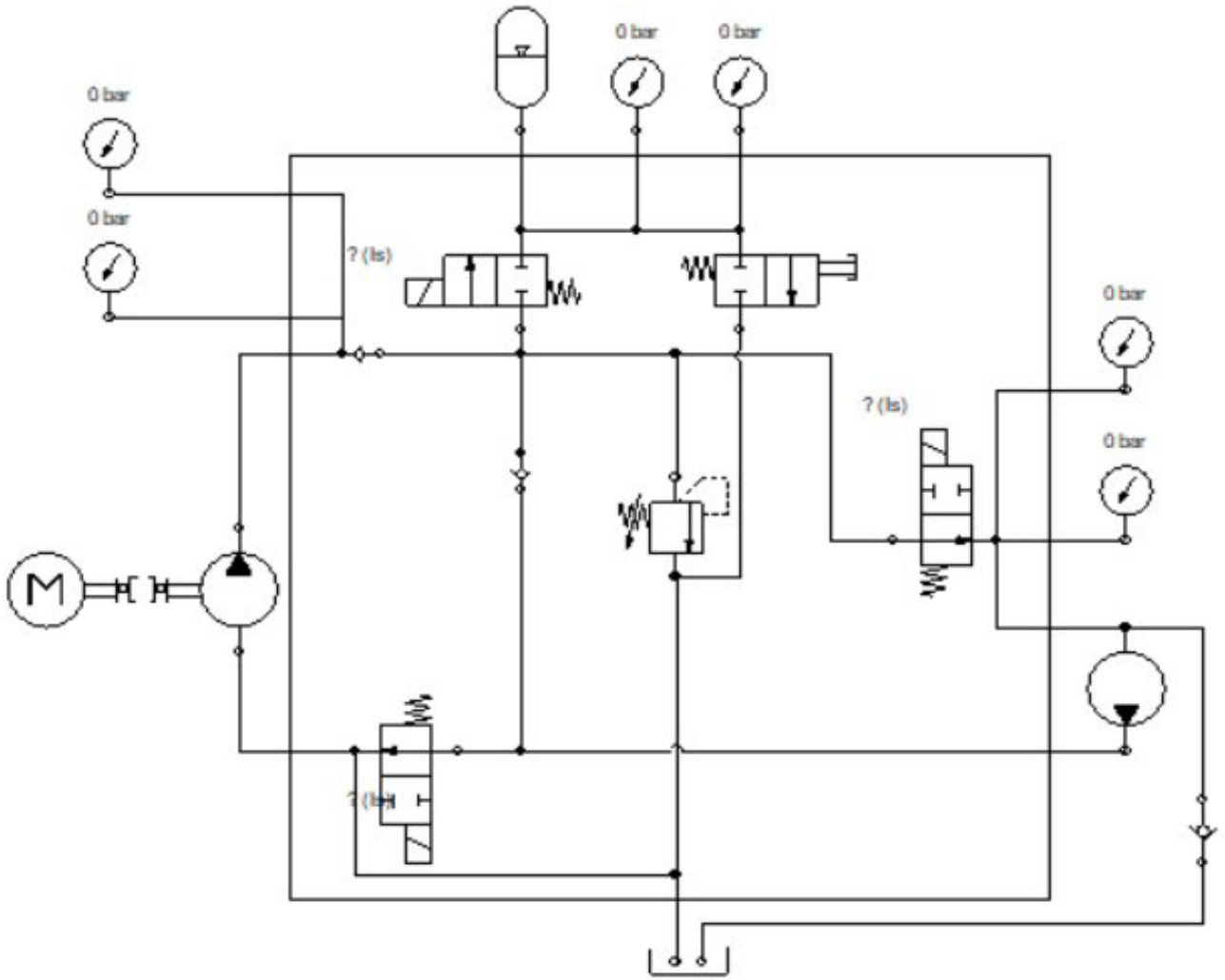
- Creates cleaner look to bike with less components

# New Components

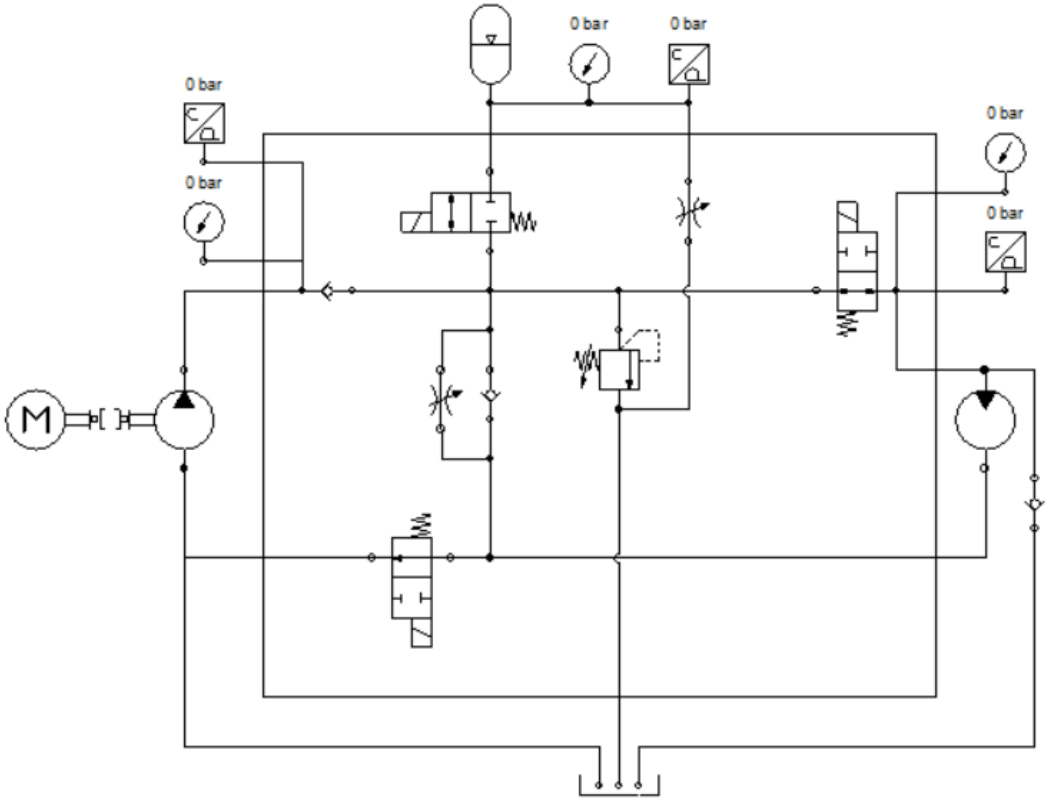
- Simplified manifold
  - ✓ Needle valve for reverse mode
  - ✓ Needle valve for smoother pressure dump

- New full usage of last year's Pinion 6-Speed Transmission
- New pump
- New motor

# Manifold: Old Circuit

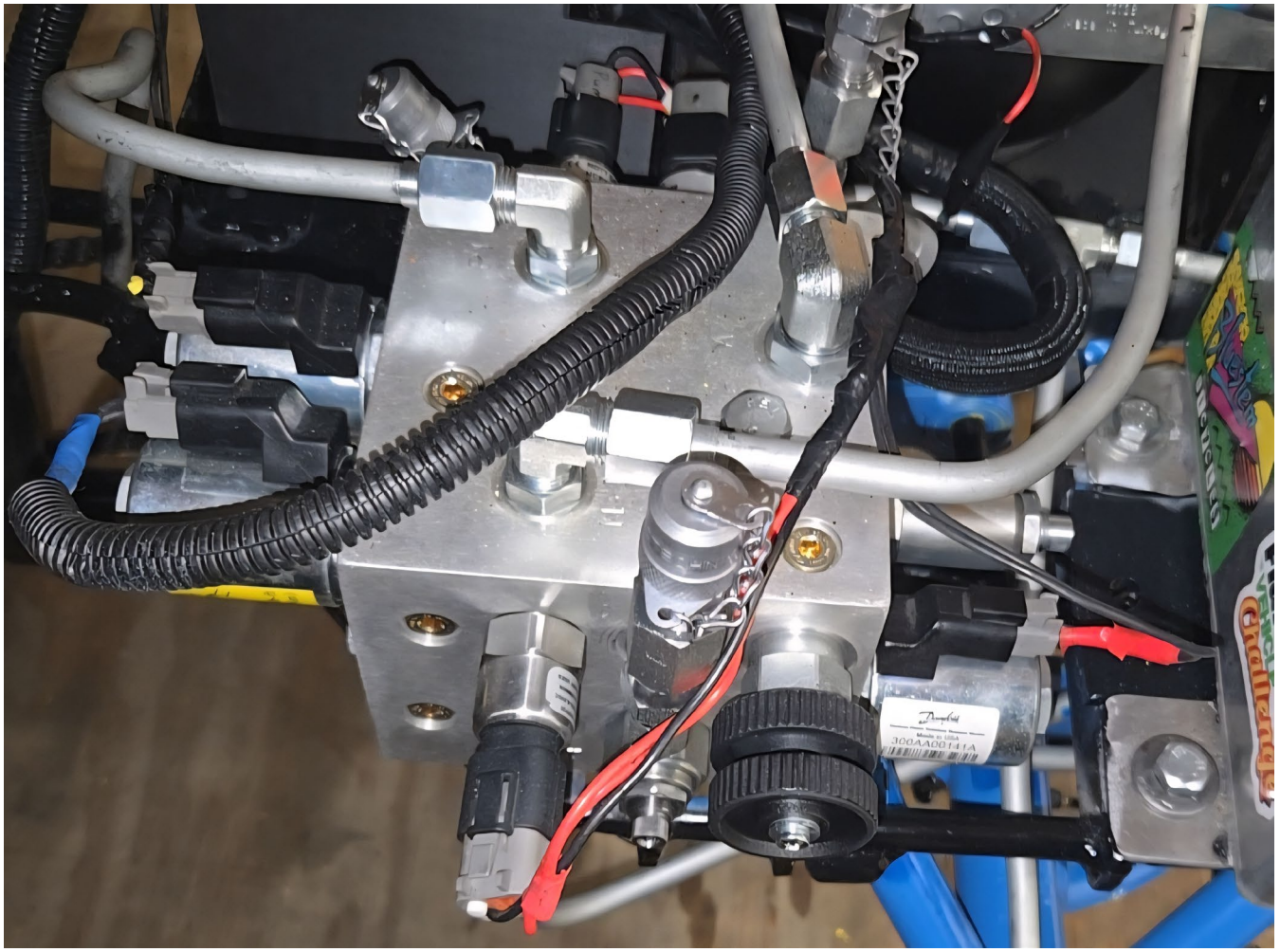


# Manifold: New Circuit

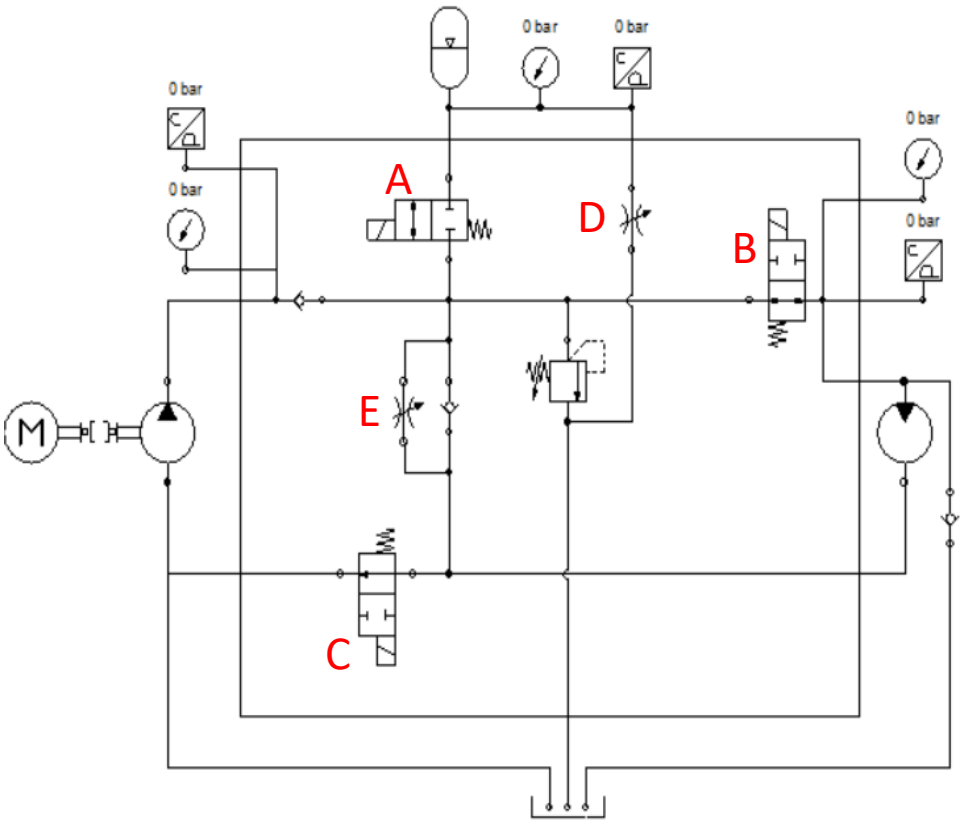



- Test ports and pressure transducers
  - Located at every component
  - Manual test ports
- Two new adjustable needle valves
  - Reverse mode
  - Smoother pressure dump

# Manifold: New Manifold



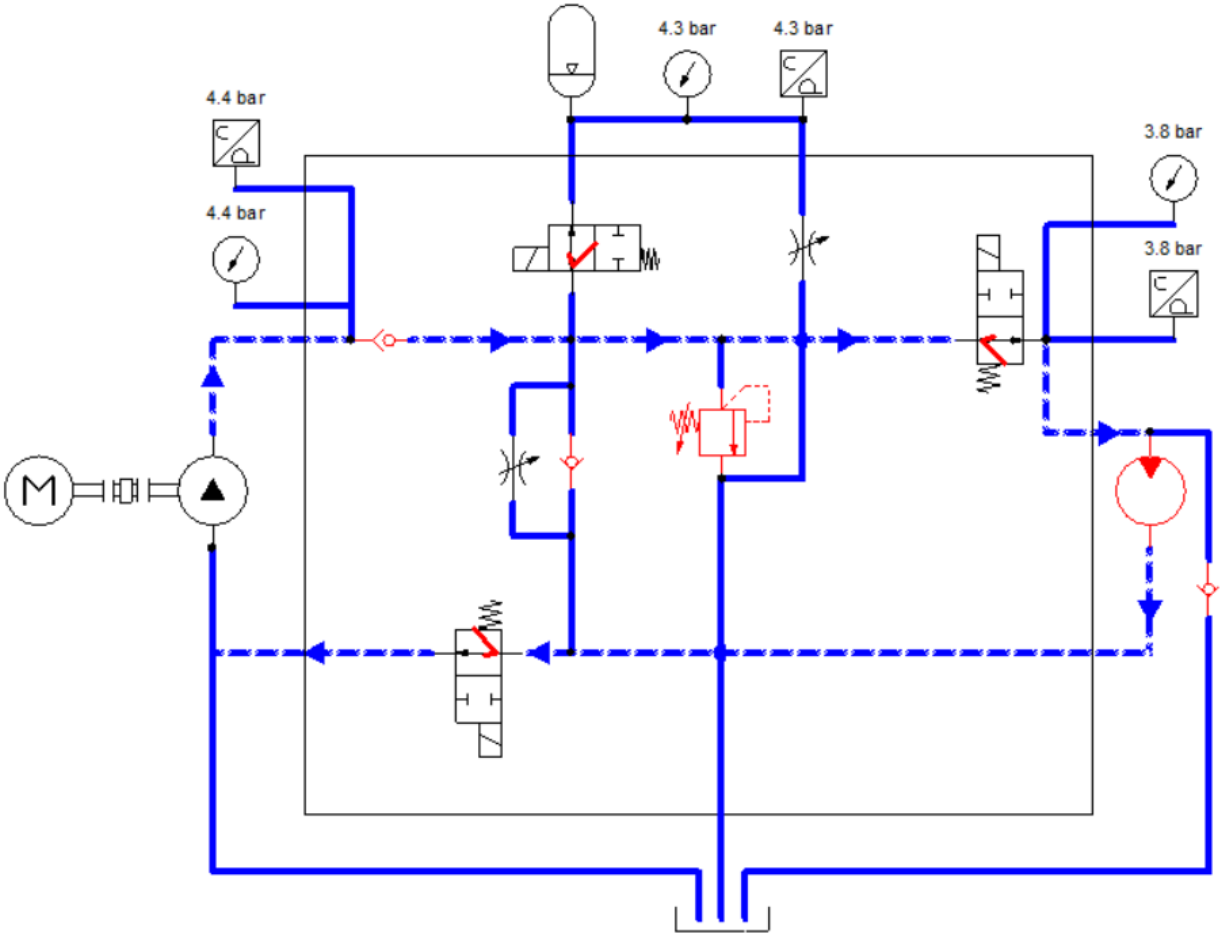
# Manifold: Valve Positions




| Mode | Valve | | | | |
|---|---|---|---|---|---|
| | A | B | C | D | E |
| Pedal to Power | 0 | 0 | 0 | 0 | 0 |
| Accumulator Charge | 1 | 1 | 0 | 0 | 0 |
| Regenerative Charge | 1 | 1 | 1 | 0 | 0 |
| Accumulator Discharge | 1 | 0 | 0 | 0 | 0 |
| System Dump | 0 | 0 | 0 | 1 | 0 |
| Reverse | 0 | 0 | 0 | 0 | 1 |

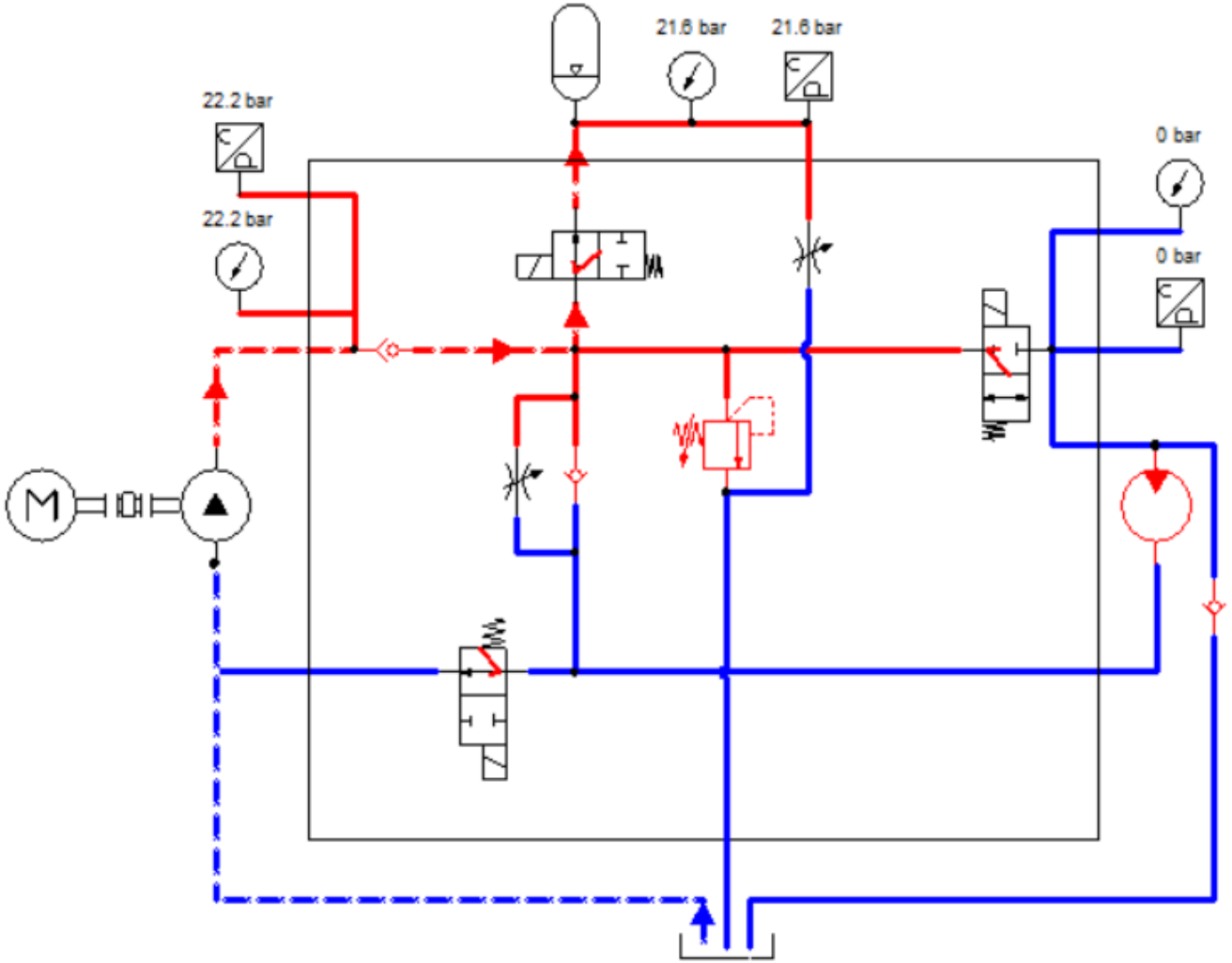
* Valves D & E are for manually operated

# Manifold: Pedal to Power



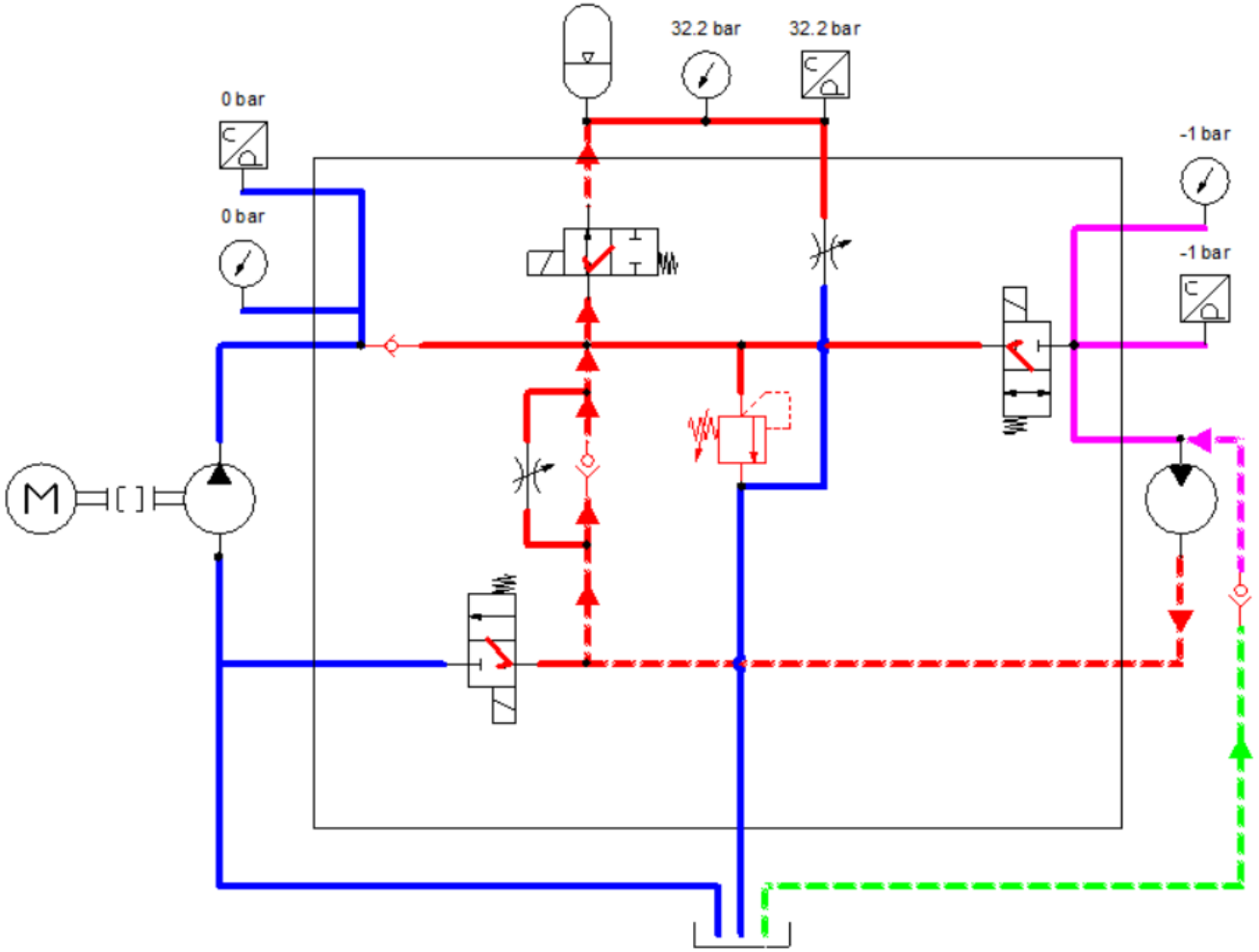

Free flow between the pump and motor allow us to pedal and move like a normal bicycle.

# Manifold: Accumulator Charge



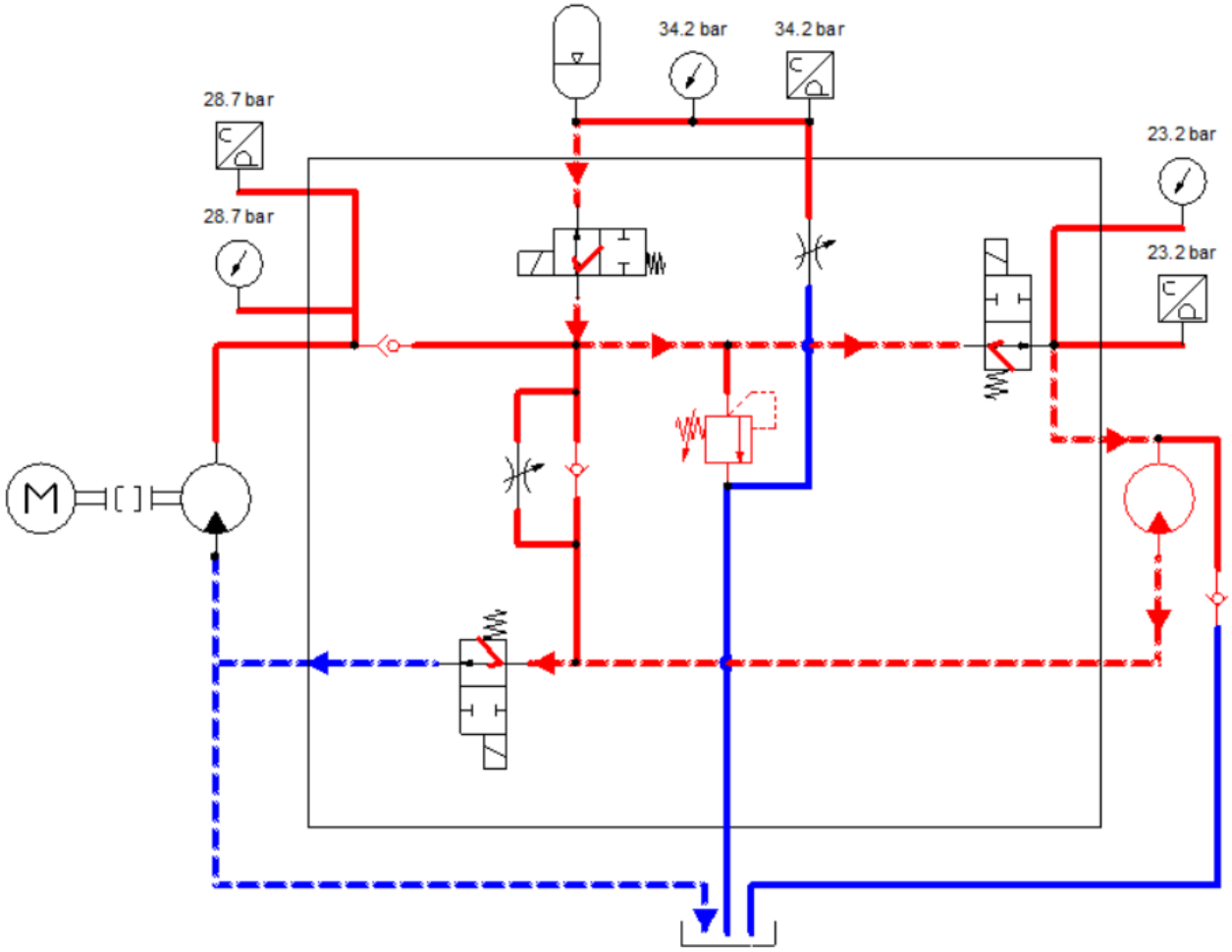

Accumulator charge allows us to pedal and build pressure into the accumulator, so we can discharge it in accumulator discharge mode.

# Manifold: Regenerative Charge



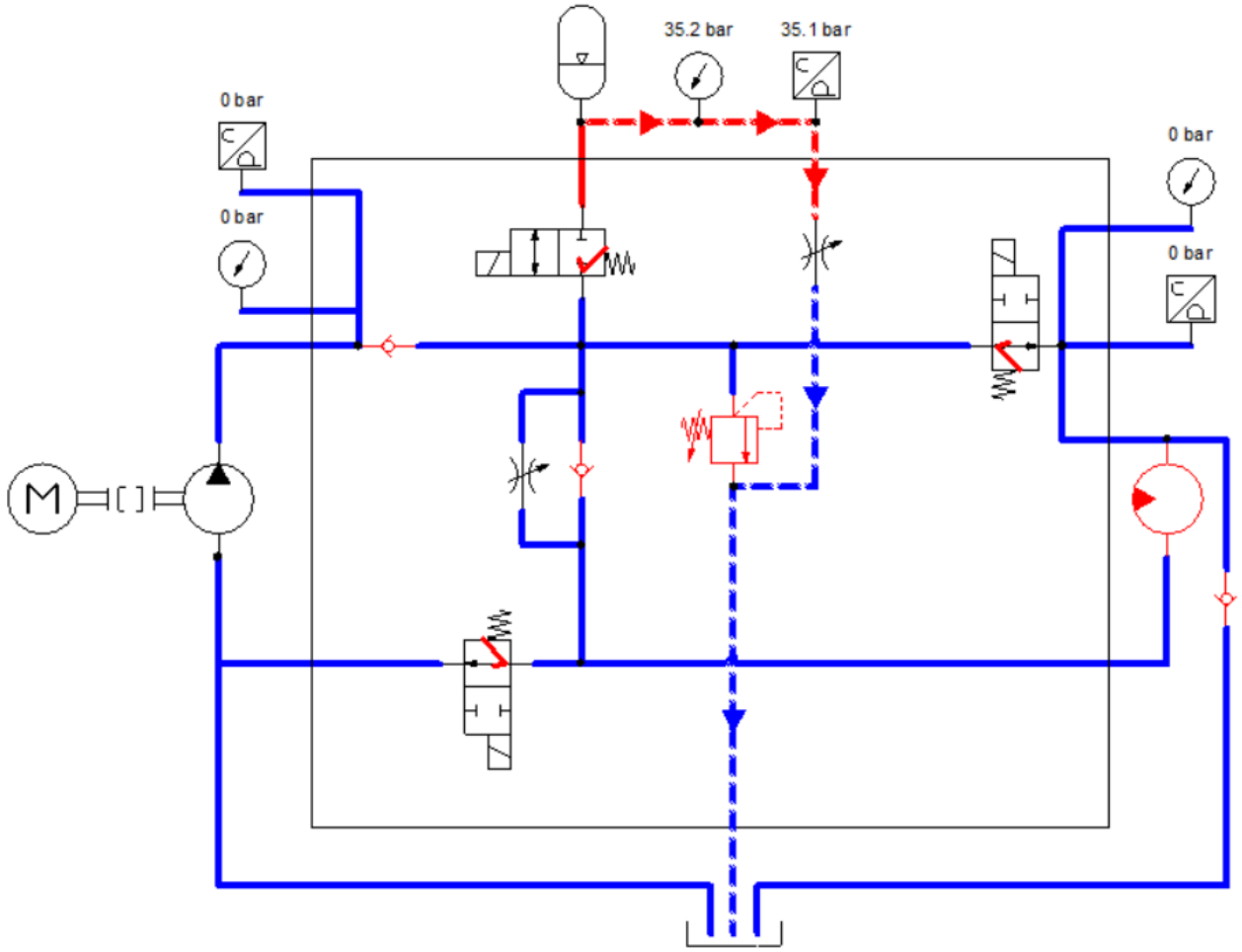

Regenerative charge mode uses the motion from the wheels to drive the motor and build up pressure in the accumulator without having to pedal. This can be used to propel us further using the speed we gained from discharging our accumulator.

# Manifold: Accumulator Discharge



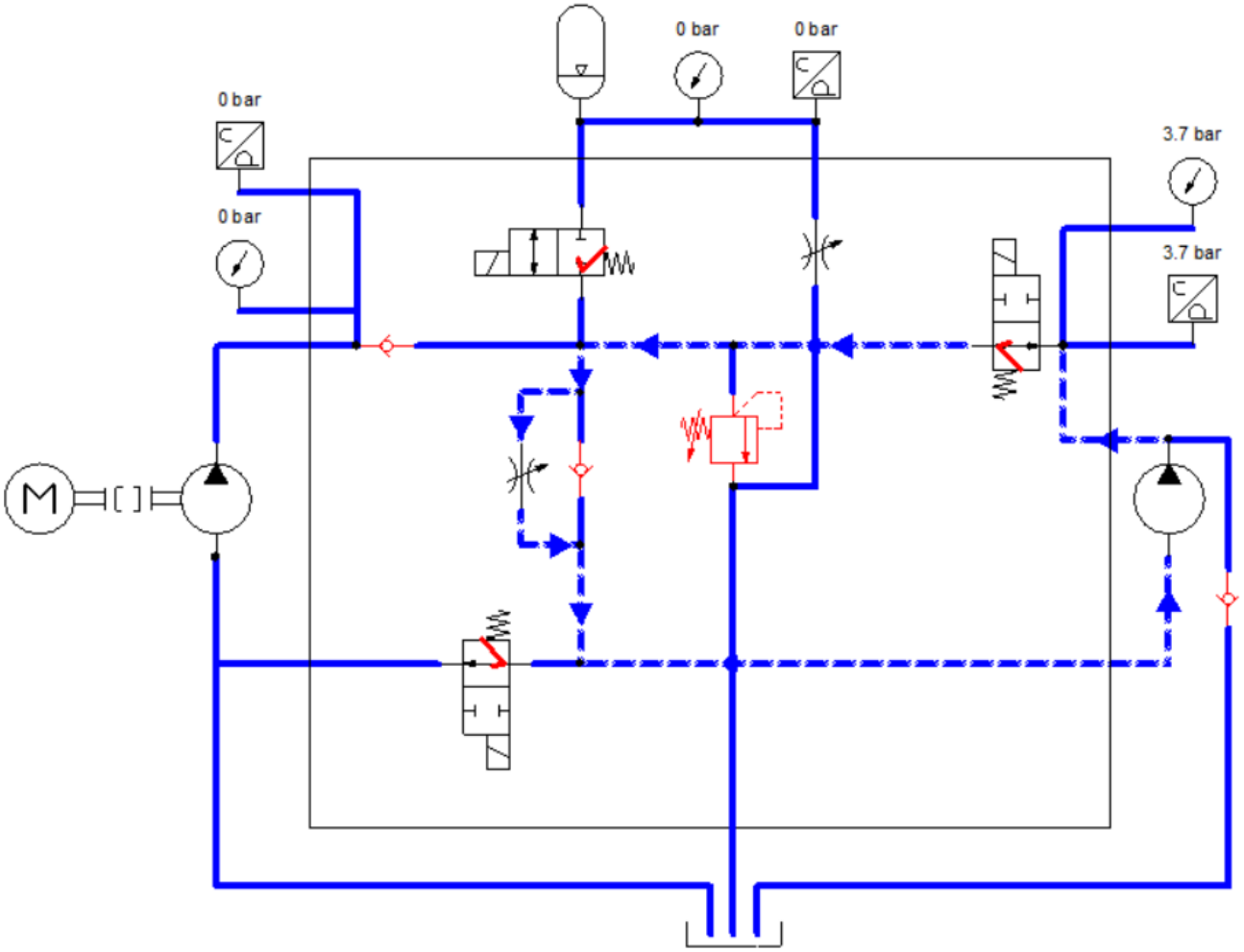

After charging the accumulator, this mode allows us to use that built-up pressure to drive the motor, propelling the bike forward.

# Manifold: Accumulator Dump




We use this valve in order to dump our accumulator pressure because it allows us to adjust the diameter of the orifice and control flow through the tubing, making for a much smoother dump than using a DCV.

# Manifold: Reverse




This valve allows flow to flow freely if we rotate the wheels in the reverse direction. In past circuits, pushing the bike in reverse would back up pressure into the check valve there and not allow us to go any further.

# Hardware Selection

## Pinion 6-speed Transmission

Current bike has low speed gear
- Good for take off
- Difficult to reach high speeds

New Transmission
- Bike will have 6 gears

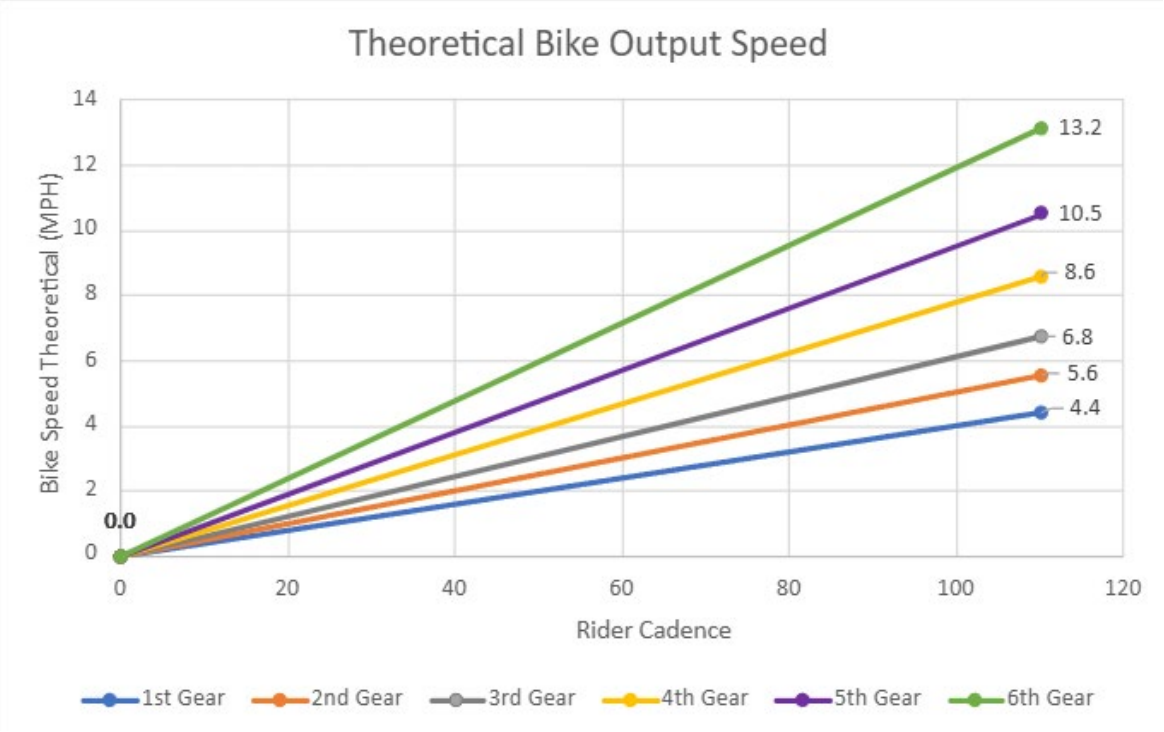

| | |
|---|---|
| Number of gears | 6 |
| Total range | 295 % |
| Step size | 24.3 % |
| Easiest gear | 0.95 |
| Fastest Gear | 0.32 |
| Gearbox weight | ca. 1800 g |

Gearbox Output Speed * (Input Gear/Output Gear)= Output Gear RPM

| P1.6 | Gearbox Ratio | gearbox output speed RPM | Pump Rpm | motor rpm | rear axle rpm | ground speed ft/min | rider input rpm | mph |
|---|---|---|---|---|---|---|---|---|
| 1 | 1.05 | 84 | 336 | 152.41 | 57.15 | 388.64 | 80.00 | 4.42 |
| 2 | 1.32 | 105.6 | 422.4 | 191.60 | 71.85 | 488.58 | 80.00 | 5.55 |
| 3 | 1.61 | 128.8 | 515.2 | 233.69 | 87.64 | 595.92 | 80.00 | 6.77 |
| 4 | 2.04 | 163.2 | 652.8 | 296.11 | 111.04 | 755.08 | 80.00 | 8.58 |
| 5 | 2.5 | 200 | 800 | 362.88 | 136.08 | 925.34 | 80.00 | 10.52 |
| 6 | 3.13 | 250.4 | 1001.6 | 454.33 | 170.37 | 1158.53 | 80.00 | 13.17 |

# New Hardware Improvements

- Reduced Operator Felt Impulse (Hydro Leduc minimum speed 200RPM, Danfoss 700RPM)
- Torque Increase
- Increased Momentum



# New Pump and Motor

**Type: Gear Pump**
**Manufacturer: Marzocchi**
**Weight: 5 lbs**
**Clockwise Rotation**
**Displacement: 2.8 $cm^3$**
**Flow Rate: 1.33 GPM**

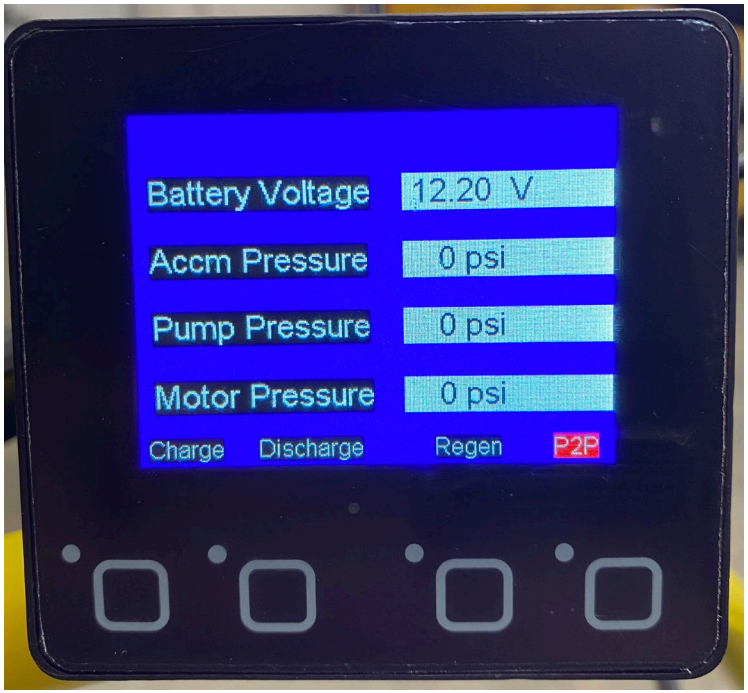
**Type: Gear Motor**
**Manufacturer: Dynamic FC**
**Weight: 7.78 lbs**
**Displacement: 5.1 $cm^3$**

# Display

- Last year's display was simple and worked very well so we decided not to change anything on it

- Reorganized wiring
  - New electronic storage box design

- MRS MConn Mini
  - Series Smallest HMI-Display
  - 2.4-inch Colored Display
  - Customizable, Waterproof, Flexible

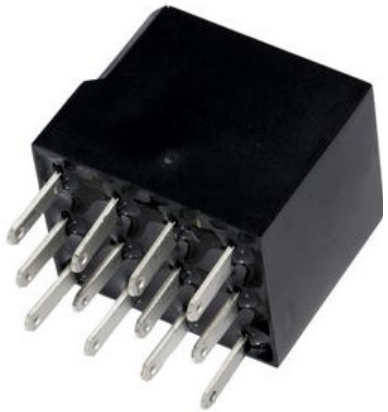



**Focus on simplicity, functionality, and easily accessible support**

# Electronic Controller

- Worked great so same as last year

- MicroPlex 7X – CAN Controller with High Integration
  - 7 inputs/outputs & 3 configurable analog or digital inputs



# Splitting Data

```
// We need this because we are sending data values that are larger than 256   We need up to 5000

//Return value of most signigicant hex bit of an int between the value 256 - 65535
int user_return_hex_MSB(int value)
{
    int MSB = (value >> 8) & 0xFF;
    return MSB;
}

//Return value of least signigicant hex bit of an int between the value 256 - 65535

int user_return_hex_LSB(int value)
{
    int LSB = (value & 0xFF);
    return LSB;
}
```

# Micro Plex Programming


```
// 3 Input/Output for the 3 coils.
user_Coil1 = can_db_get_value(0, Coil_1);          //Read CAN Message
os_pin_write(Coil1, user_Coil1);                   //set a Digital Output

user_Coil2 = can_db_get_value(0, Coil_2);          //Read CAN Message
os_pin_write(Coil2, user_Coil2);                   //set a Digital Output

user_Coil3 = can_db_get_value(0, Coil_3);          //Read CAN Message
os_pin_write(Coil3, user_Coil3);                   //set a Digital Output

os_can_send_message(0x720, 0, 8, user_heartbeat, 0, 0, 0, 0, 0, 0, 0, 0);                  //Sending CAN Heartbeat Message
// os_can_send_message(0x101, 0, 8, user_Coil1, user_Coil2, user_Coil3, 0, 0, 0, 0, 0);     //Sending CAN Message for 3 coils..... They can be sent individually in PCAN View  //101 = Coil_Messag
//Andrew said to comment out line above

// 3 Gauge Readings and the Battery Voltage.

user_BattVolt =  os_algin_mv(Batt_Volt);             // Reading Battery Voltage          (Read Analog Input)   (Name of Pin)      Battery Voltage =       Pin 2
user_AccmPress = os_algin_mv(Accm_Press);            // Reading Accumulator Presssure    (Read Analog Input)   (Name of Pin)   Accumulator Pressure =  Pin 8
user_PumpPress = os_algin_mv(Pump_Press);            // Reading Pump Presssure           (Read Analog Input)   (Name of Pin)   Pump Pressure =         Pin 7
user_MotorPress = os_algin_mv(Motor_Press);          // Reading Motor Presssure          (Read Analog Input)   (Name of Pin)  Motor Pressure =        Pin 9

tmpbyte_batt_volt_MSB = user_return_hex_MSB(user_BattVolt);
tmpbyte_batt_volt_LSB = user_return_hex_LSB(user_BattVolt);

tmpbyte_accm_press_MSB = user_return_hex_MSB(user_AccmPress);
tmpbyte_accm_press_LSB = user_return_hex_LSB(user_AccmPress);

tmpbyte_pump_press_MSB = user_return_hex_MSB(user_PumpPress);
tmpbyte_pump_press_LSB = user_return_hex_LSB(user_PumpPress);

tmpbyte_motor_press_MSB = user_return_hex_MSB(user_MotorPress);
tmpbyte_motor_press_LSB = user_return_hex_LSB(user_MotorPress);
```

# Display Static Code

```
// clear display once and prepare the static content of the display page and
if(flag_clear_do_once == 0)
{
    flag_clear_do_once = 1;
    // Prepare background, clear old content
    board_set_lcd_background_color(LCD_COLOR_BLUE );

    // print the gauge reading information Titles
    board_lcd_print_text( 10  , 40    , "Battery Voltage", LCD_FONT_ARIAL_22, LCD_COLOR_WHITE, LCD_COLOR_BLACK);
    board_lcd_print_text( 10  , 86    , "Accm Pressure", LCD_FONT_ARIAL_22, LCD_COLOR_WHITE, LCD_COLOR_BLACK);
    board_lcd_print_text( 10  , 132   , "Pump Pressure", LCD_FONT_ARIAL_22, LCD_COLOR_WHITE, LCD_COLOR_BLACK);
    board_lcd_print_text( 10  , 180   , "Motor Pressure", LCD_FONT_ARIAL_22, LCD_COLOR_WHITE, LCD_COLOR_BLACK);

}
```

# Display Status Code

```
//Mode Determination for 4 Buttons
//Pedal to Power
if ((user_Coil1_State == 0) && (user_Coil2_State == 0) && (user_Coil3_State == 0))
{
    board_lcd_print_text( 280 , 220, "P2P", LCD_FONT_ARIAL_16, LCD_COLOR_WHITE, LCD_COLOR_RED);
    board_lcd_print_text( 5 , 220, "Charge", LCD_FONT_ARIAL_16, LCD_COLOR_WHITE, LCD_COLOR_BLACK);
    board_lcd_print_text( 200   , 220, "Regen", LCD_FONT_ARIAL_16, LCD_COLOR_WHITE, LCD_COLOR_BLACK);
    board_lcd_print_text( 80   , 220, "Discharge", LCD_FONT_ARIAL_16, LCD_COLOR_WHITE, LCD_COLOR_BLACK);
}
//Accumulator Charge
else if((user_Coil1_State == 1) && (user_Coil2_State == 1) && (user_Coil3_State == 0))
{
    board_lcd_print_text( 280 , 220, "P2P", LCD_FONT_ARIAL_16, LCD_COLOR_WHITE, LCD_COLOR_BLACK);
    board_lcd_print_text( 5 , 220, "Charge", LCD_FONT_ARIAL_16, LCD_COLOR_WHITE, LCD_COLOR_RED);
    board_lcd_print_text( 200   , 220, "Regen", LCD_FONT_ARIAL_16, LCD_COLOR_WHITE, LCD_COLOR_BLACK);
    board_lcd_print_text( 80   , 220, "Discharge", LCD_FONT_ARIAL_16, LCD_COLOR_WHITE, LCD_COLOR_BLACK);
}
//Mode Determination for Regen
else if((user_Coil1_State == 1) && (user_Coil2_State == 1) && (user_Coil3_State == 1))
{
    board_lcd_print_text( 280 , 220, "P2P", LCD_FONT_ARIAL_16, LCD_COLOR_WHITE, LCD_COLOR_BLACK);
    board_lcd_print_text( 5 , 220, "Charge", LCD_FONT_ARIAL_16, LCD_COLOR_WHITE, LCD_COLOR_BLACK);
    board_lcd_print_text( 200   , 220, "Regen", LCD_FONT_ARIAL_16, LCD_COLOR_WHITE, LCD_COLOR_RED);
    board_lcd_print_text( 80   , 220, "Discharge", LCD_FONT_ARIAL_16, LCD_COLOR_WHITE, LCD_COLOR_BLACK);
}
//Accumulator Discharge
else if((user_Coil1_State == 1) && (user_Coil2_State == 0) && (user_Coil3_State == 0))
{
    board_lcd_print_text( 280 , 220, "P2P", LCD_FONT_ARIAL_16, LCD_COLOR_WHITE, LCD_COLOR_BLACK);
    board_lcd_print_text( 5 , 220, "Charge", LCD_FONT_ARIAL_16, LCD_COLOR_WHITE, LCD_COLOR_BLACK);
    board_lcd_print_text( 200   , 220, "Regen", LCD_FONT_ARIAL_16, LCD_COLOR_WHITE, LCD_COLOR_BLACK);
    board_lcd_print_text( 80   , 220, "Discharge", LCD_FONT_ARIAL_16, LCD_COLOR_WHITE, LCD_COLOR_RED);
}
else
{
    board_lcd_print_text( 280 , 220, "P2P", LCD_FONT_ARIAL_16, LCD_COLOR_WHITE, LCD_COLOR_RED);
    board_lcd_print_text( 5   , 220, "Charge", LCD_FONT_ARIAL_16, LCD_COLOR_WHITE, LCD_COLOR_RED);
    board_lcd_print_text( 200 , 220, "Regen", LCD_FONT_ARIAL_16, LCD_COLOR_WHITE, LCD_COLOR_RED);
    board_lcd_print_text( 80  , 220, "Discharge", LCD_FONT_ARIAL_16, LCD_COLOR_WHITE, LCD_COLOR_RED);
}
```

# Testing:

- Accumulator pre-charged to 800psi
- Pulsing discharge for increased distance
- Adjusting pressure relief valve to just under 3,000 psi limit
- 50-yard speed testing (old vs new)

| OLD BIKE | |
|---|---|
| Person A Time (sec) | Person B Time (sec) |
| 12.91 | 12.86 |
| 13.83 | 13.05 |
| 15.26 | 13.26 |

| NEW BIKE | | |
|---|---|---|
| Person A Time (sec) | Person B Time (sec) | Strategy |
| 10.92 | 11.14 | Shifting |
| 11.02 | 11.31 | Shifting |
| 11.14 | 11.28 | 3rd Gear |

# Lessons Learned

- Hydraulic design basics
- Hydraulic fittings and tubing
- Accumulator – charging, valve, etc.
- Plumbing
- Time management
- Basic teamwork skills
- Communication skills
- Troubleshooting

# Questions?