r001-lineFollowerBasic

Rev :: 2024-09-19

:|:|: General :|:|:

= Deliverables =

Frist

1. Scematic (A3 size, landscape; by hand ok)
2. Complete component documentation (organized directory with pn as name and readme)
3. Initial BOM -- for testing (excel)

Second

1. Schematic (A3 size, landscape)
2. Working prototype -- soldered christmas tree
3. Document outlining: (markdown or latex)

Third

1. Schematic (A3 size, landscape)
2. Complete BOM
3. Design documents
4. Proyect documentation
5. Gerber files

Fourth

1. Approved documentation

:|:|: Project :|:|:

Overview

This project involves designing a Printed Circuit Board (PCB) for a line follower robot. The PCB will include all necessary electronic components and subsystems to control and power the robot, focusing on sensor integration, motor control, and microcontroller implementation. This project does not include the mechanical components of the robot but centers on the electronics. Students are expected to design a custom PCB that interfaces effectively with the robot's hardware and meets the outlined specifications.

Subsystems and Components

1. **Sensor Array Subsystem:**
    - **Design Requirements:**
        - Incorporate a sensor bar with 16 receiver-emitter LED pairs for line detection.
        - Resistor values for the emitter and receiver LEDs may differ. Select resistors carefully to ensure optimal operation of both the emitter and receiver.
        - Use appropriate current-limiting resistors to protect the LEDs and ensure consistent performance.
        - Make sure the receiver LEDs are sensitive enough to detect the contrast between the line and the background (typically black and white surfaces).
    - **Data Handling:**
        - Connect the outputs of the LED sensors to an analog multiplexer. The purpose of the multiplexer is to reduce the number of analog inputs needed on the microcontroller.
        - The analog input pin of the microcontroller should be connected to the output of the multiplexer. Each input of the multiplexer will be connected to the respective LED receiver's positive pin (or another suitable point in the circuit).
        - Implement a reset mechanism to ensure the multiplexer starts reading from sensor 1 and continues in sequence. Consider using a dedicated reset pin or a software-based counter to track the sensor position.
2. **Motor Control Subsystem:**
    - **H-Bridge Configuration:**
        - Use two H-Bridge circuits built with MOSFETs to control the two DC motors. These H-Bridges will allow for both direction and speed control of the motors.
        - When driving the motors in a given direction, consider having one MOSFET fully on (for continuous current flow) while the other MOSFET is pulse-width modulated (PWM) for speed control. Alternatively, both MOSFETs can be PWM-controlled, but this might introduce additional complexity in switching times and heat dissipation.

- **Protection and Control:**
  - Implement diodes across the motor terminals to prevent back electromotive force (EMF) from damaging the MOSFETs.
  - Include resistors on the MOSFET gates to control the switching speed and avoid ringing.
  - Consider adding capacitors to smooth voltage fluctuations caused by motor switching. Place a low-ESR capacitor near the motor connections to suppress noise and protect the circuit from transients.
  - Include motor connectors, such as Mini-Fit Jr., and ensure proper overcurrent protection with fuses. A fuse for each motor can prevent damage in the event of excessive current draw.
  - Implement current sensing for each motor using a shunt resistor and an INA219 sensor. To handle the PWM signal, place the shunt resistor in the low-side configuration between the MOSFET and ground. This allows for accurate current sensing, though some filtering may be necessary to handle the PWM switching noise.
- **Encoder Input:**
  - Provide connectors for encoders on each motor to allow for precise speed control. The encoders will feed back the motor's actual speed to the microcontroller, which can use this data for closed-loop control to maintain consistent speeds.

3. **Microcontroller Unit (MCU):**
   - **Core Requirements:**
     - The microcontroller should be based on the ATmega328P, the same processor used in the Arduino Uno. This will ensure compatibility with the Arduino IDE.
     - Include essential components such as a 16 MHz crystal oscillator for stable timing and clock generation. Make sure to add the required coupling capacitors (typically 22pF) to ensure proper oscillator operation.
     - Use bypass capacitors (100nF) near the MCU's VCC and GND pins to reduce noise and ensure stable power delivery.
     - For bootstrapping the custom Arduino board, use the Arduino IDE with the correct board settings for an ATmega328P. Ensure the correct bootloader is installed, allowing the MCU to be programmed via a serial interface with an FTDI chip.
     - **Reset Button:** Include a reset button on the PCB for manually resetting the MCU. This can be helpful for debugging and restarting the system.
   - **Additional Features:**
     - Integrate a 6-pin, dual-row 0.1" male header for programming the board using an FTDI USB-to-serial adapter. The header should connect to the RX, TX, VCC, GND,

DTR, and RESET pins of the MCU. Ensure that the RESET pin is properly connected to allow for automatic resetting during programming.

4. **Power Management:**
   - **Battery System:**
     - The system will use eight AA-sized NiMH rechargeable batteries (1.2V each) for a total voltage of 9.6V. These batteries will power the motors directly through the H-Bridge circuits.
     - Ensure the battery connector is compatible with an 8-bay NiMH battery holder.
   - **Voltage Regulation:**
     - Use a low-dropout (LDO) voltage regulator to step down the battery voltage (9.6V) to 5V for powering the microcontroller and sensors. A linear regulator like the LM7805 could be used, but for better efficiency, especially considering battery-powered operation, a switching regulator (buck converter) is recommended.
     - To ensure clean power for sensitive analog circuits (such as the sensor array and MCU), use filtering capacitors (e.g., 10µF electrolytic and 100nF ceramic) on the output of the regulator to reduce noise.
     - Place an appropriate decoupling capacitor (e.g., 100nF) near the analog input pins to stabilize the analog readings from the sensor array.
5. **User Interface and Additional Components:**
   - **Accelerometer and Gyroscope:**
     - Integrate an accelerometer and gyroscope into the design to enable inertial navigation. These sensors will provide feedback on the robot's orientation and movement, which can be used to enhance line following or for additional functionality like obstacle avoidance. Consider using a combined IMU (Inertial Measurement Unit) such as the MPU-6050, which provides both accelerometer and gyroscope data over I2C.
   - **Indicators and Controls:**
     - Include an LED on pin 13 for standard Arduino functionality.
     - Add 5 multipurpose LEDs for general-purpose signaling, providing visual feedback during robot operation.
     - Design the board with 3 multipurpose buttons for user interaction and a DSP switch with 5 selectors for mode selection.
   - **Connectivity and Safety:**
     - Provide an 8-bay NiMH battery holder for power and a removable fuse holder for easy replacement. Include a fuse for overall system protection.
     - Integrate a master on/off switch to control the robot's power without having to disconnect the battery.

General Considerations

- **Testing and Validation:**
  - Outline procedures to test each subsystem, particularly focusing on sensor responsiveness, motor control, current sensing, and power management. Test the PCB in stages to ensure that each component functions as intended before full system integration.
  - Implement a basic test program to ensure that the microcontroller can read from the sensor array, control motor speed and direction, and respond to user input from the buttons and switches.

Conclusion

This project challenges students to design and implement a custom PCB for a line follower robot. It requires careful planning of sensor integration, motor control, and microcontroller operation, providing a valuable learning experience in electronic design, PCB layout, and embedded systems. By following this specification, students will gain hands-on experience with real-world design challenges while creating a functional and adaptable platform for future robotic projects.

:|:|: Resources :|:|:

General resources

- Design: https://easyeda.com
- https://www.flux.ai

Other resources

- https://www.instructables.com/Arduino-IDE-Creating-Custom-Boards/
- https://www.instructables.com/DIY-Arduino-UNO-How-to-Make-Your-Own-Arduino-Uno-B/
- https://www.youtube.com/watch?v=1hN74CKubtQ