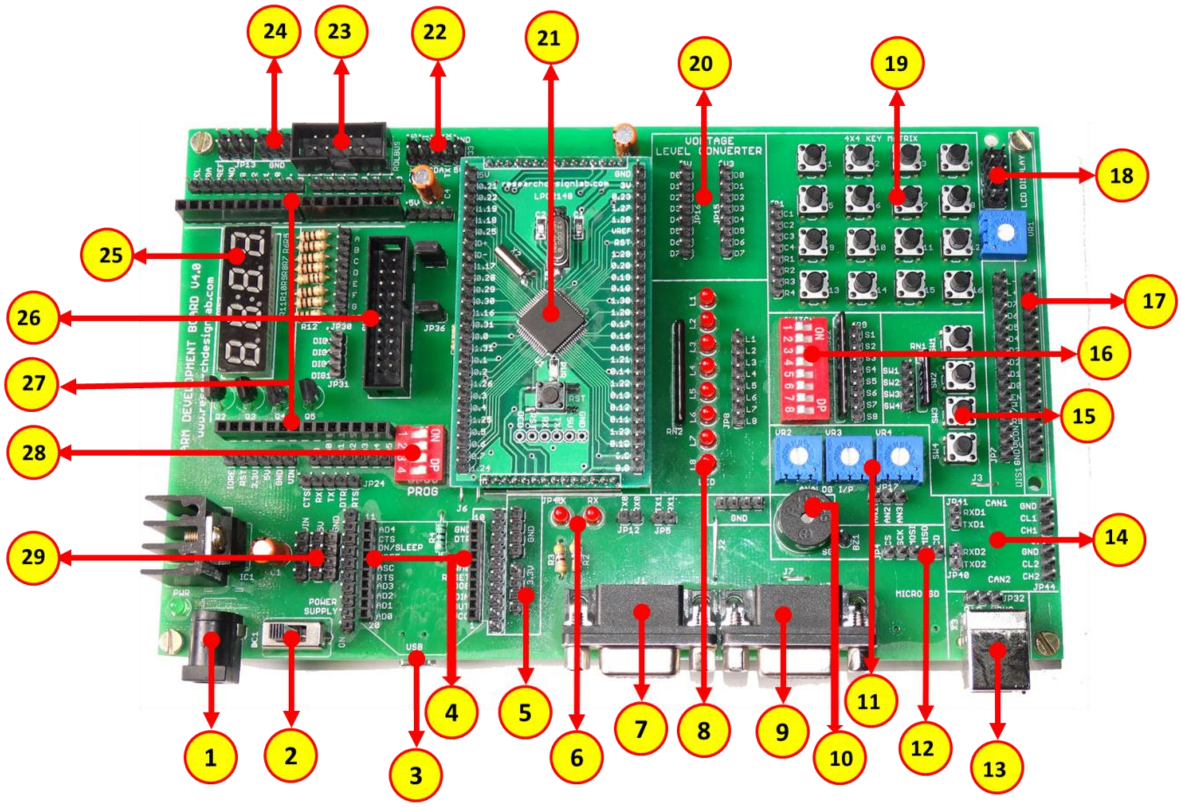
ARM DEVELOPMENT BOARD LPC2129-TRAINER KIT

RDL742

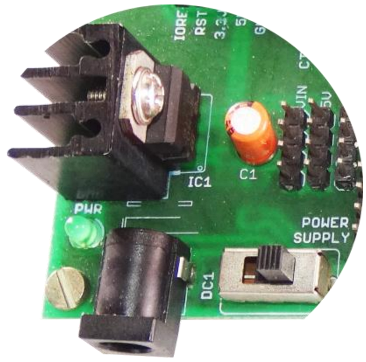


1. Power supply.
2. Power ON switch
3. USB Programming port **
4. XBEE footprint/ XBEE Adaptor module
5. DC 3.3V connectors
6. TX and RX LED's
7. DB-9 serial female connector-1
8. 8x1 LED's
9. DB-9 serial female connector-2
10. Buzzer
11. Variable resistor POT
12. Micro SD memory card connector**
13. ARM in built USB port
14. CAN bus*
15. 4x1 Keypad
16. 8 way DIP switch
17. 16x2 LCD connectors
18. Node connector
19. 4x4 Keypad matrix
20. Bi-Directional Voltage Level Converter (3.3V-5V)
21. ARM LPC2129
22. RDL Bus Connector
23. RDL Bus FRC Connector
24. GND Pin Outs
25. 4x1 7 Segment display
26. JTAG Programmer
27. Stackable header for Arduino Shields.
28. Program/Run Mode selector
29. DC 12V, 5V, GND connectors.

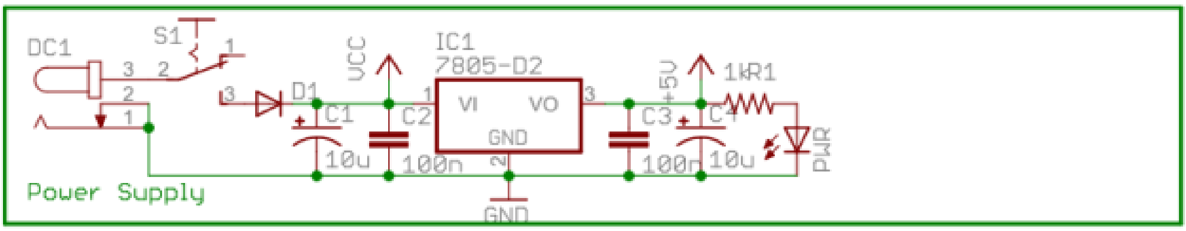
*Optional

** SD card holder and mini USB port are placed at the bottom of the PCB.

Power supply, 5V-12V



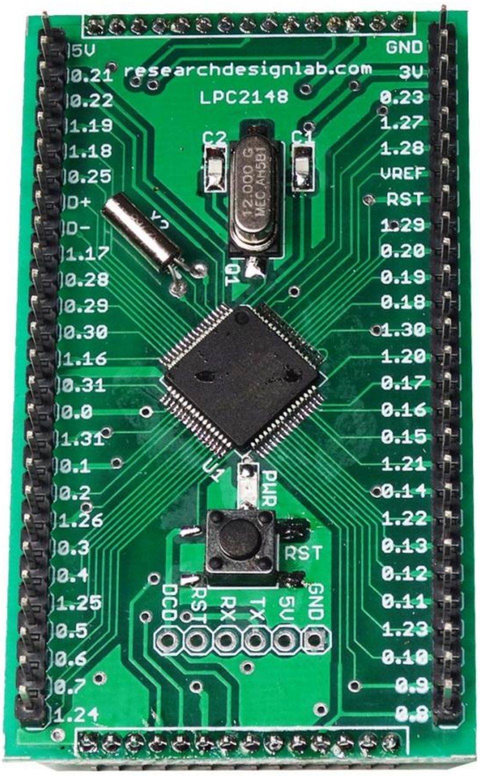
All digital circuits require regulated power supply. Here is a simple power supply circuit diagram used on this board. You can use AC or DC source (12V) which converts into regulated 5V which is required for driving the

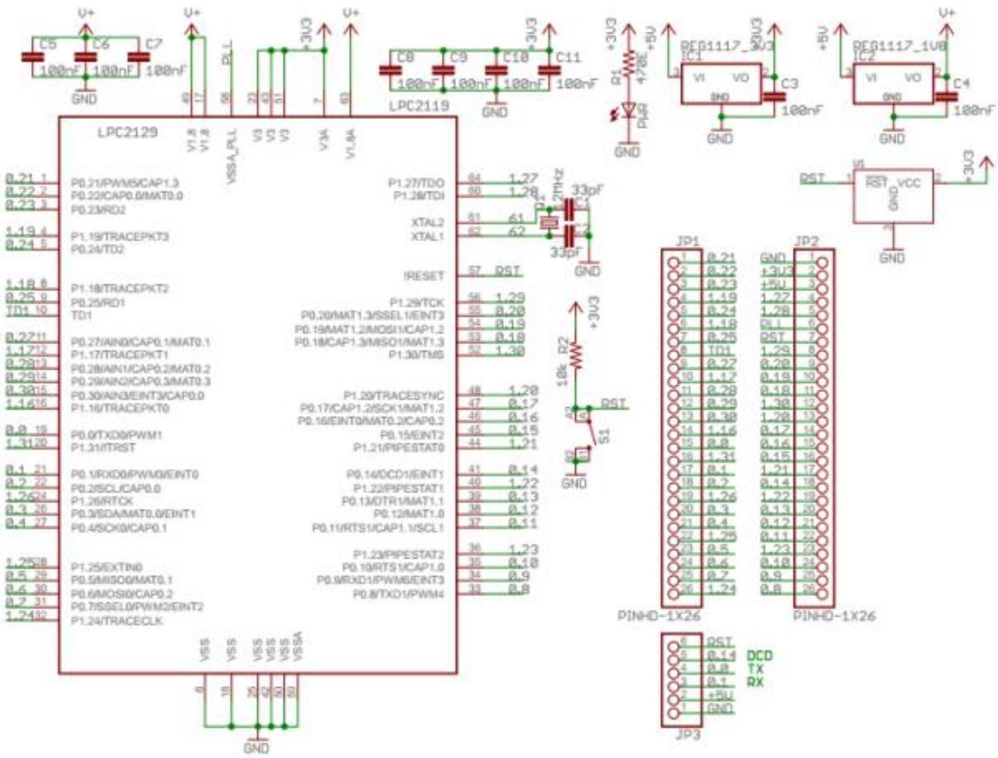


development board circuit.

ARM LPC2129

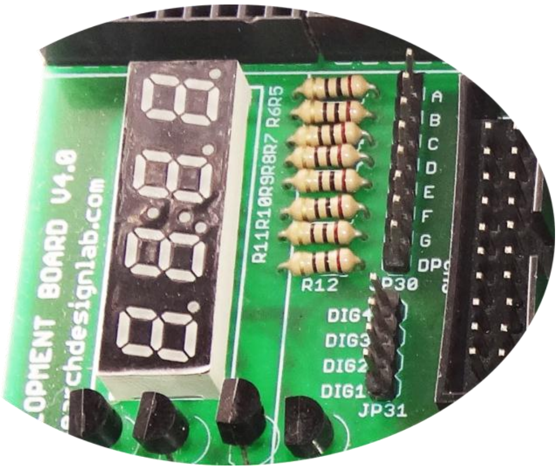
Female stackable header for mounting various 52 pin ARM processors.

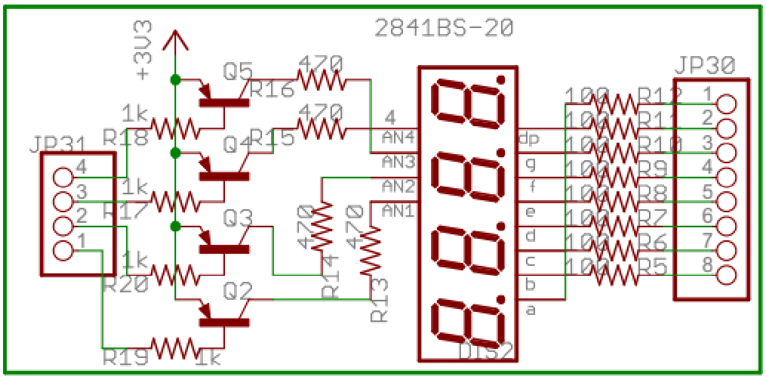




4 Digit 7 Segment Display

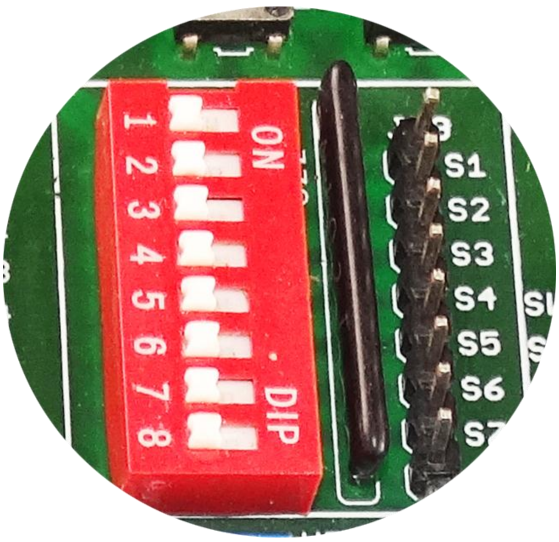
One seven segment digit consists of 7+1 LEDs which are arranged in a specific formation which can be used to represent digits from 0 to 9 and even some letters. One additional LED is used for marking the decimal dot, in case you want to write a decimal point in the desired segment.

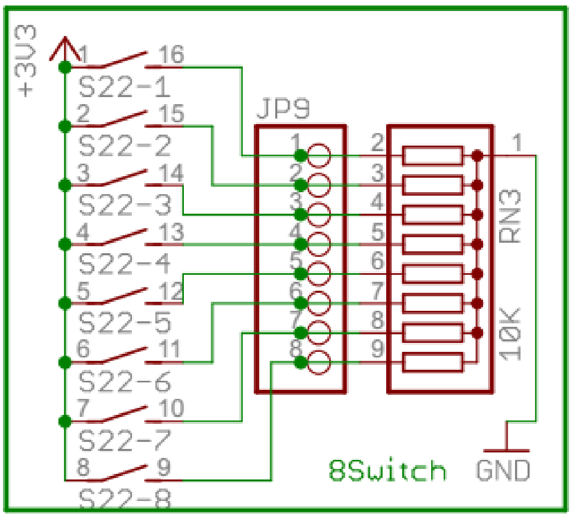




8 Way DIP Switch

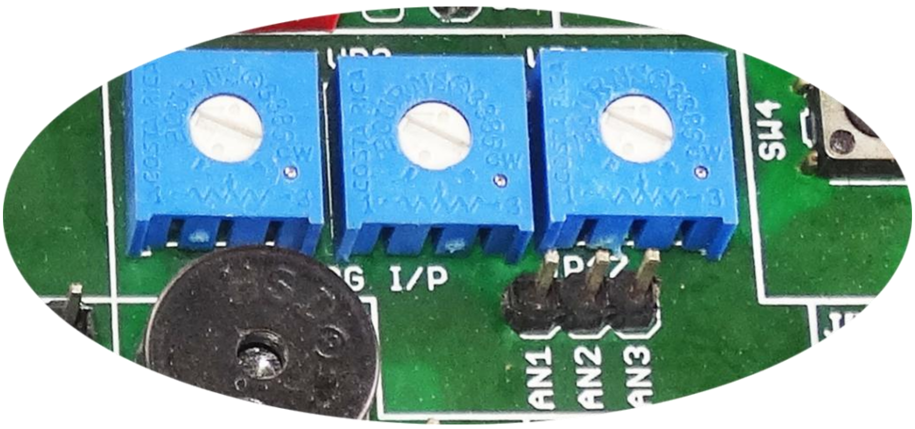
DIP switches are an alternative to jumper blocks. Their main advantages are that they are quicker to change and there are no parts on lose.

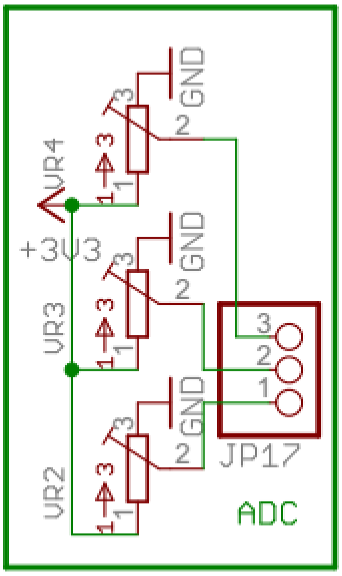




Potentiometer:

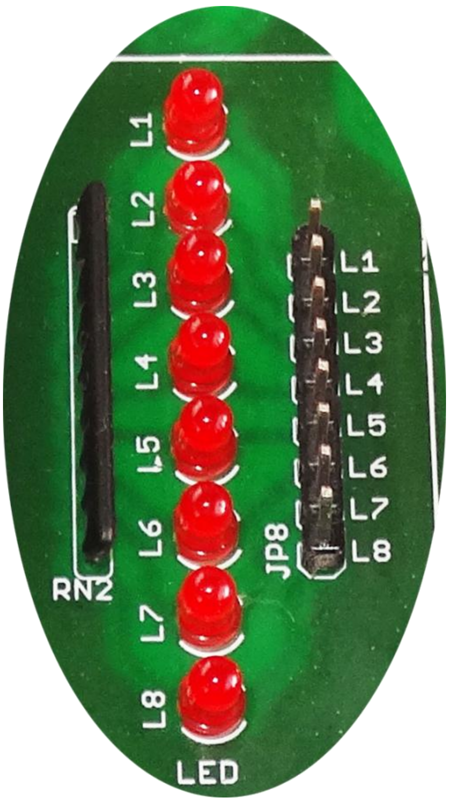
The Potentiometer Option allows the user to adjust the voltage reference by rotating a potentiometers dial. Turning the potentiometer changes the voltage reference making it easier to adjust the motor speed and also to set the duty cycle for PWM values(via programming).

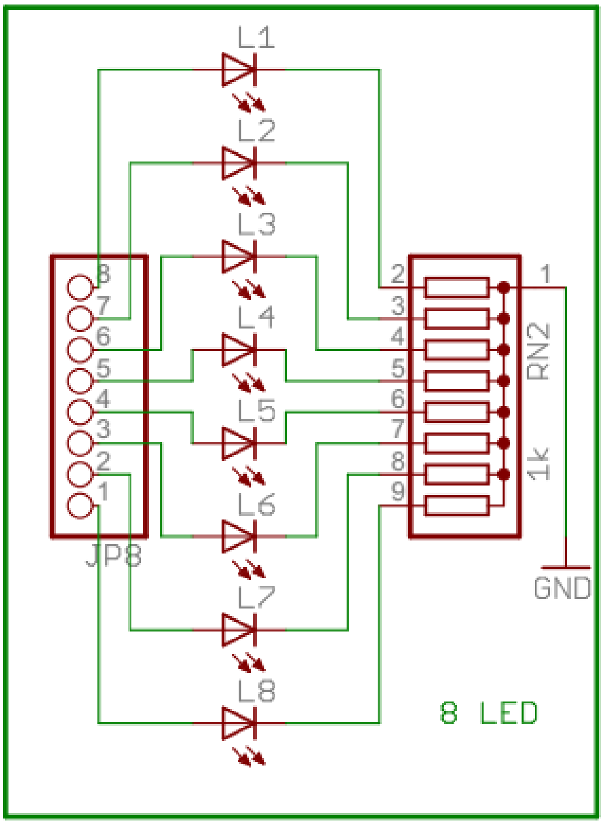




8x1 LED's

LED's are used to indicate something, whether any pin is high or indicating the output for many purposes like indicating I/O status or program debugging running state. We have 8 led outputs on board which can be used by the programmer as per the requirement for testing and development.





16x2 LCD Connectors

LCD screen consists of two lines with 16 characters each. Each character consists of 5x7 dot matrix. Contrast on display depends on the power supply voltage and whether messages are displayed in one or two lines. For that reason, variable voltage 0-Vdd is applied on pin marked as Vee. Trimmer potentiometer is usually used for that purpose. Some versions of displays have built in backlight (blue or green diodes). When used during operating, a resistor for current limitation should be used (like with any LE diode). LCD Connection Depending on how many lines are used for connection to the microcontroller, there are 8-bit and 4-bit LCD modes. The appropriate mode is determined at the beginning of the process in a phase called "initialization". In the first case, the data are transferred through outputs D0-D7 as it has been already explained. In case of 4-bit LED mode, for the sake of saving valuable I/O pins of the microcontroller, there are only 4 higher bits (D4-D7) used for communication, while other may be left unconnected.

Consequently, each data is sent to LCD in two steps: four higher bits are sent first (that normally would be sent through lines D4-D7), four lower bits are sent afterwards. With the help of initialization, LCD will correctly connect and interpret each data received. Besides, with regards to the fact that data are rarely read from LCD (data mainly are transferred from microcontroller to LCD) one more I/O pin may be saved by simple connecting R/W pin to the Ground. Such saving has its price. Even though message displaying will be normally performed, it will not be possible to read from busy flag since it is not possible to read from display.

Features:

1. Can display 224 different symbols.

2. Low power consumption.

3. 5x7 dot matrix format.

4. Powerful command set and user produced characters.

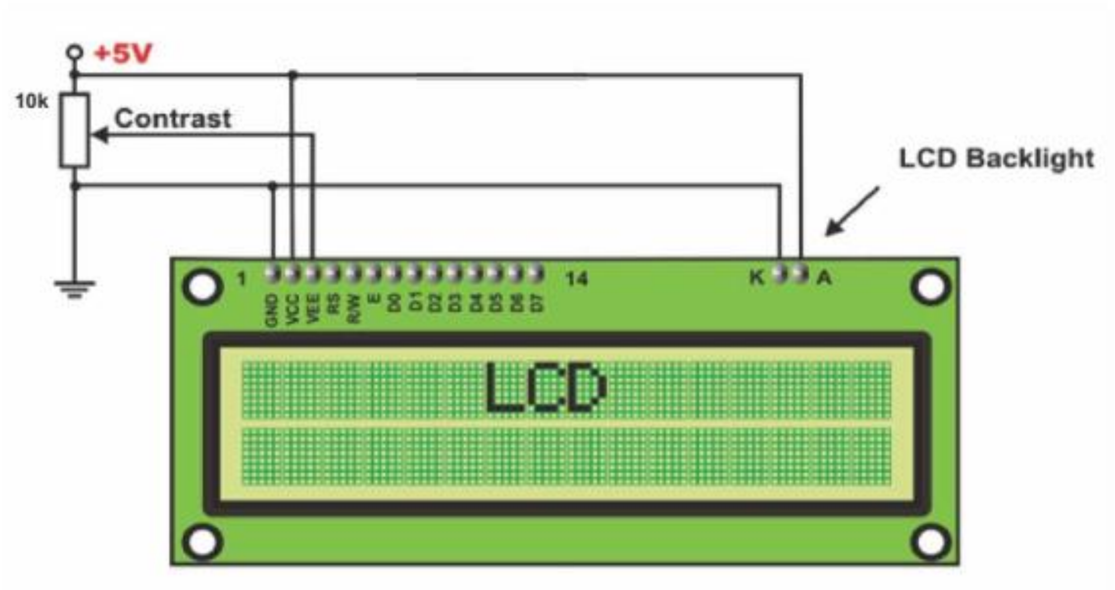
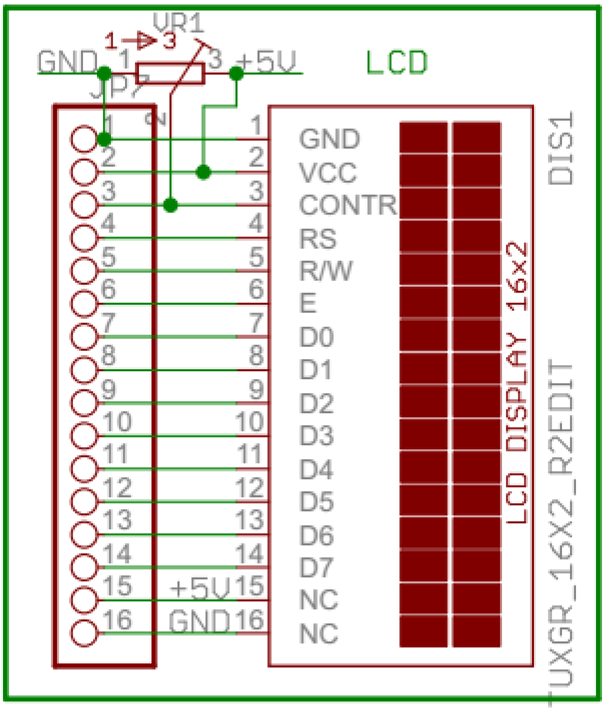


Fig: Circuit connections of LCD

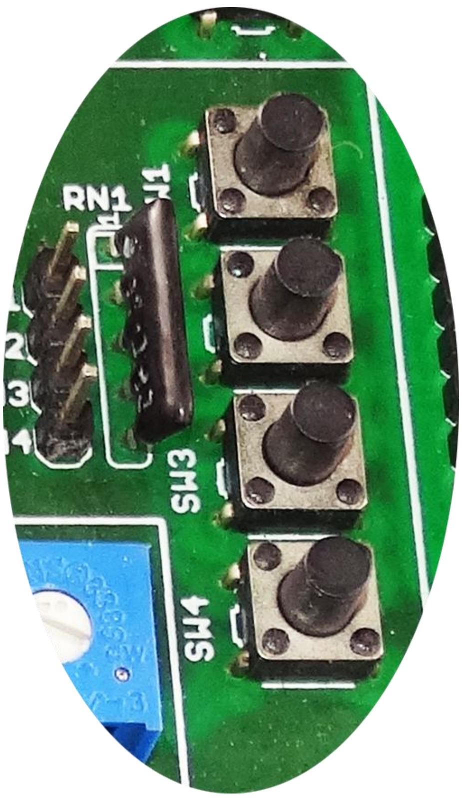


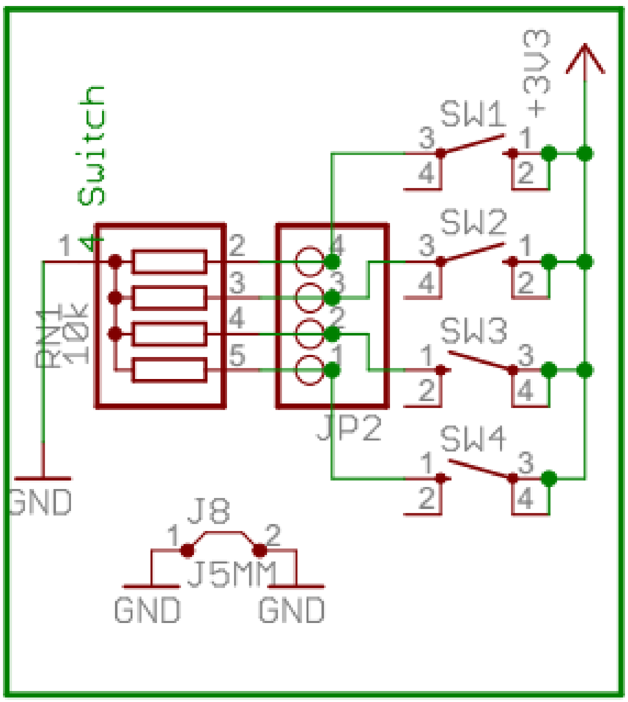
PIN DESCRIPTION

1. Gnd:- Power supply ground
2. VCC:-+5v Power supply input
3. RS:-Register Select
4. R/W:- Read/Write pin
5. En:-Enable pin
6. D0-D7:- Data lines

4x1 Keypad:

Switches are mainly used to switch the controls of a module. We have four switches on board which can be used by the programmer as per the requirement for testing and development.



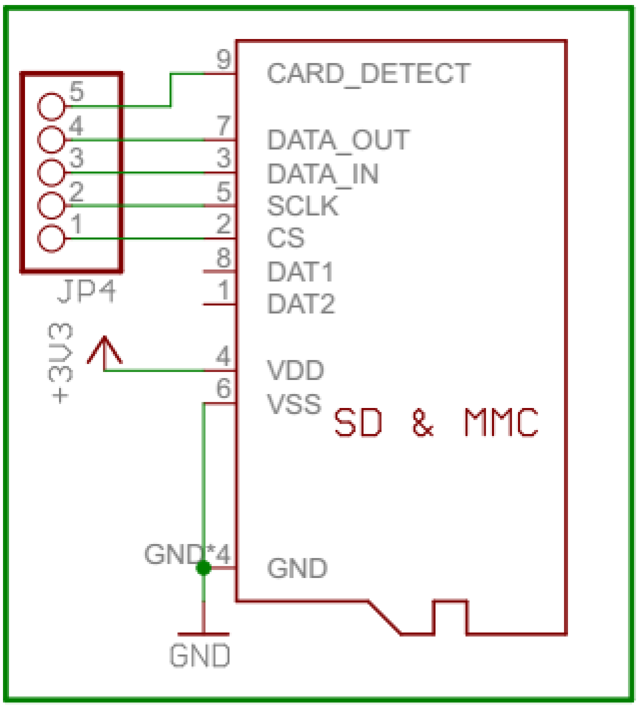


MMC/SD CARD CONNECTOR

Micro SD is currently the world's smallest memory card storage system, specifically designed for use with ultra small mobile phones and other devices. Like the mini SD, the microSD/TransFlash is ideal for use in storing media-rich files such as music, videos, and photographs in compatible mobile phones. Fujitsu's micro SD connector offers the lowest profile height (1.65mm) currently available in the market. Featuring Fujitsu's precise and insert molding technology, the FCN-560 series micro SD connector measures 5mm (D) x 4mm (W) x .6mm (H) and has a life cycle of ten thousand insertions/withdrawals. Fujitsu's connector uses a proprietary push-push structure that provides positive tactile feedback during card insertion and withdrawal. Acard drop protection mechanism prevents the card from falling out during pre-insertion and forced card extraction after it's locked in. When ejected, the card travels 3.5mm from its locked position.

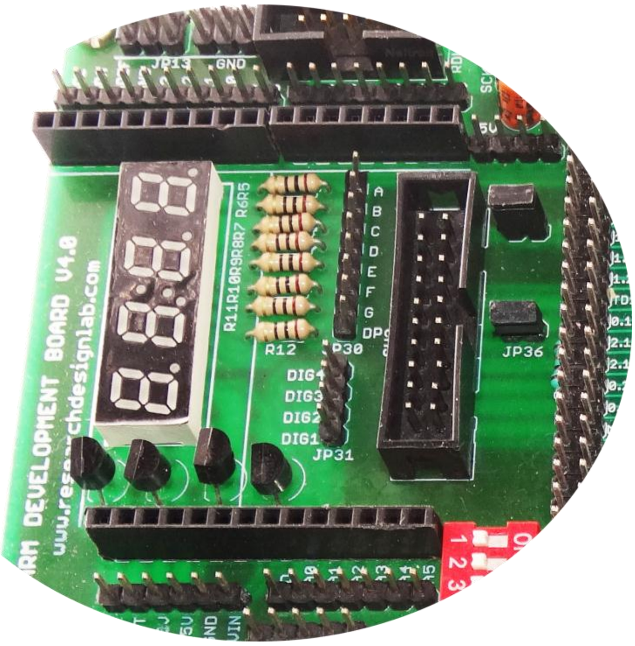
FEATURES

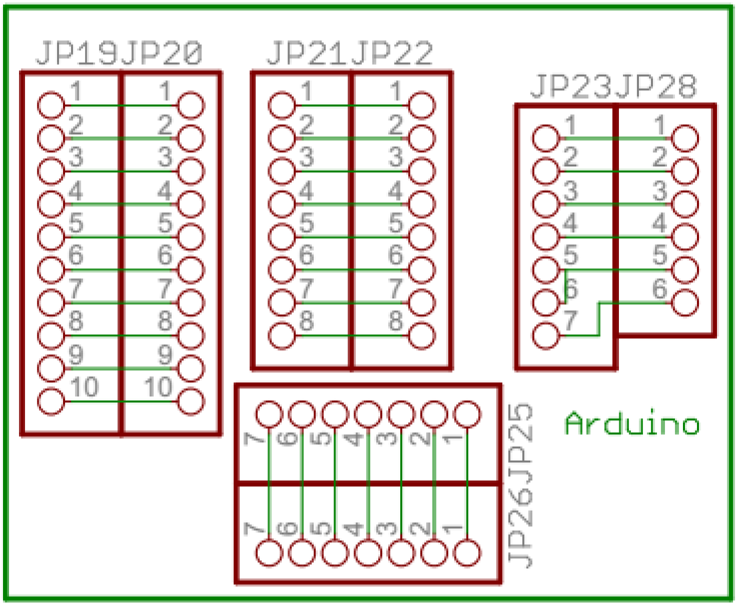
1. Reduction in size and weight (5 (D) x 4 (W) x .65 (H)mm) with unique push-push structure
2. Half-lock to prevent forced card extraction, good operability with sense of click for insertion
3. Smooth card insertion / withdrawal with unique drop protection mechanism
4. with card detections switch
5. Card ejection distance 3.5mm
6. FCN-568Z008-G/0M



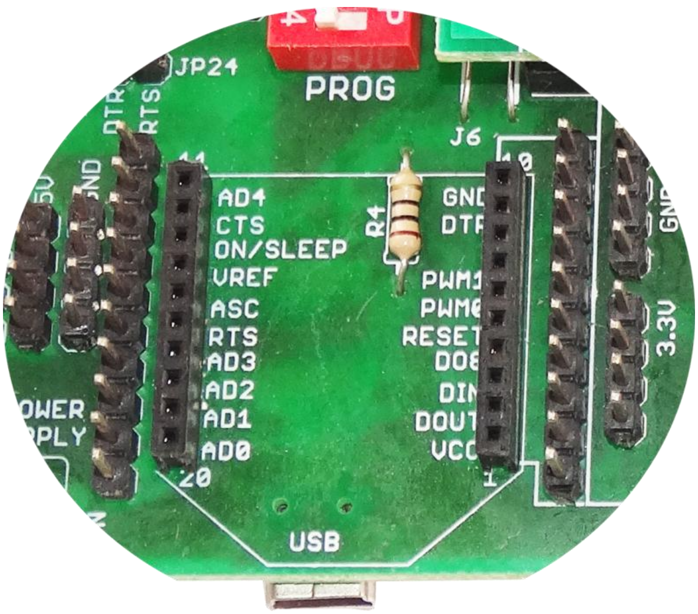
Arduino Shield Footprint

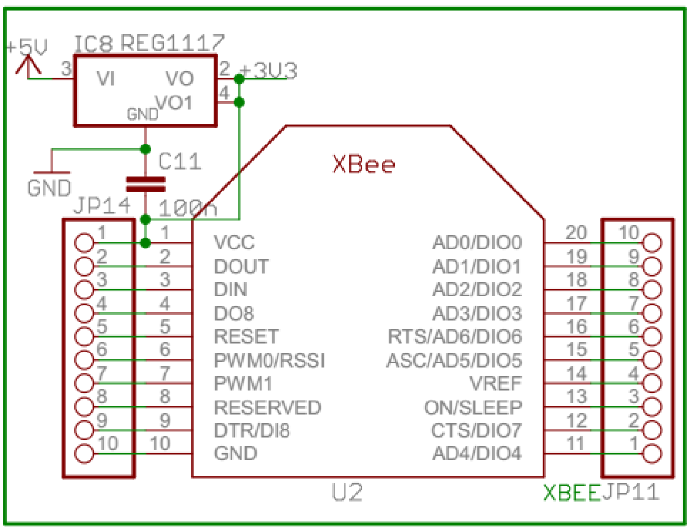
Arduino Shield footprint is provided in the board to mount different types of Arduino compatible shields on this development board.





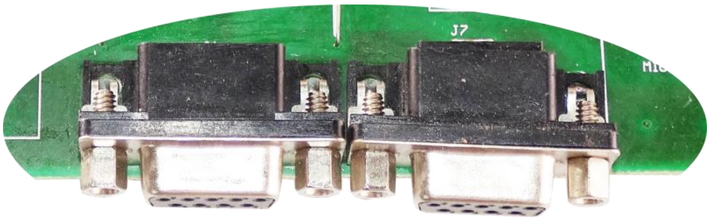
XBEE Footprint:

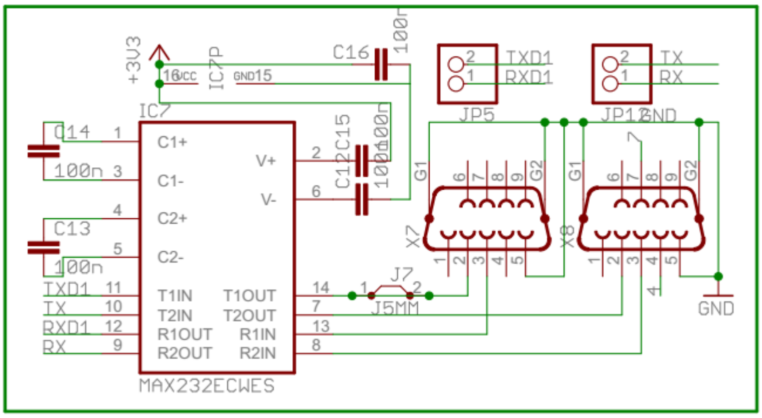




DB-9 female connectors:

R S - 2 3 2 i s a s t a n d a r d communication protocol for linking computer and its peripheral devices to allow serial data exchange. In simple terms RS232 defines the voltage for the path used for data exchange between the devices. It specifies common voltage and signal level, common pin wire configuration and minimum, amount of control signals.





4 x 4 Matrix Keypad:

In a 4x4 matrix keypad eight Input/output ports are used for interfacing with any microcontrollers. Rows are connected to Peripheral Input/output (PIO) pins configured as output. Columns are connected to PIO pins configured as input with interrupts.

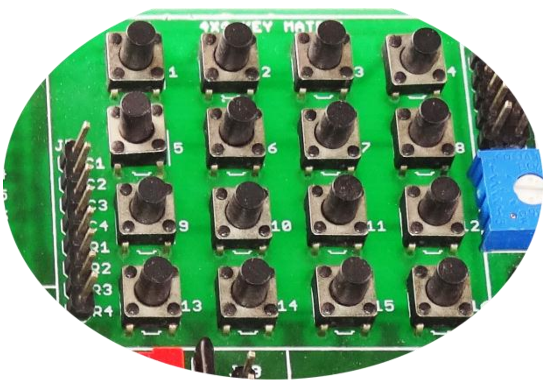
FEATURES :

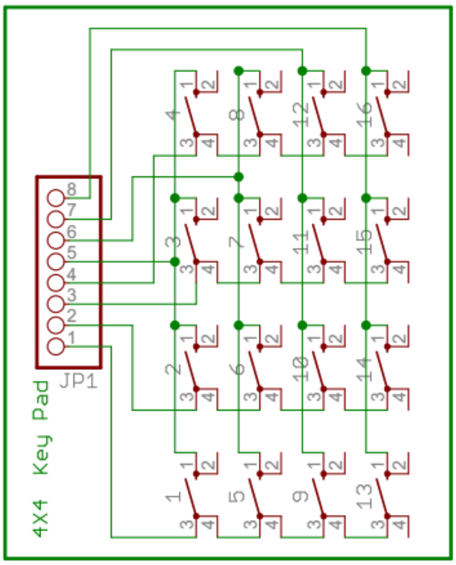
1. Contact debouncing.

2. Easy to interface.

3. Interfaces to any microcontroller or microprocessor.

4. Data valid output signal for interrupt activation.





Voltage Level Convertor:

Bidirectional Logic Level Converter is a Four-Channel Device which can be used for Voltage Level Shifting between two devices. The level converter is very easy to use. The board needs to be powered from the two voltages sources (high voltage and low voltage) that your system is using. High voltage (5V for example) to the 'H' pin, low voltage (3.3V for example) to 'L', and ground from the system to the 'GND' pin.

FEATURES:

1. Minimum Voltage: 3.3V and Maximum Voltage: 5V
2. Bi-directional Logic Level conversion is possible.
3. Bread Board friendly.

WORKING:

Bi-Directional MOSFET Voltage Level Converter when connecting 3.3V devices and 5V devices voltage level conversion is required. The following circuit will allow this to be done bi-directionally:

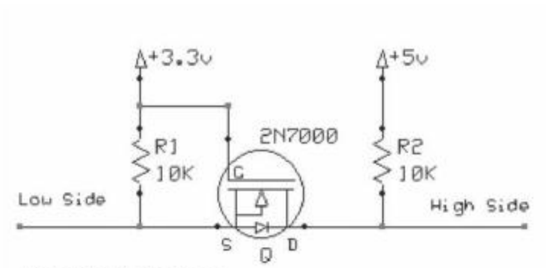


Fig: Circuit Diagram

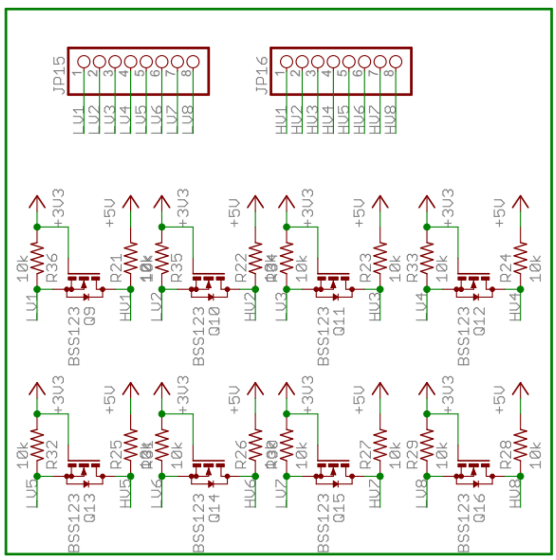
LowSideControl

When the low side (3.3V) device transmits a '1' (3.3V), the MOSFET is tied high (off), and the high side sees 5V through the R2 pull-up resistor. When the low side transmits a '0' (0V), the MOSFET source pin is grounded and the MOSFET is switched on and the high side is pulled down to 0V.

HighSideControl

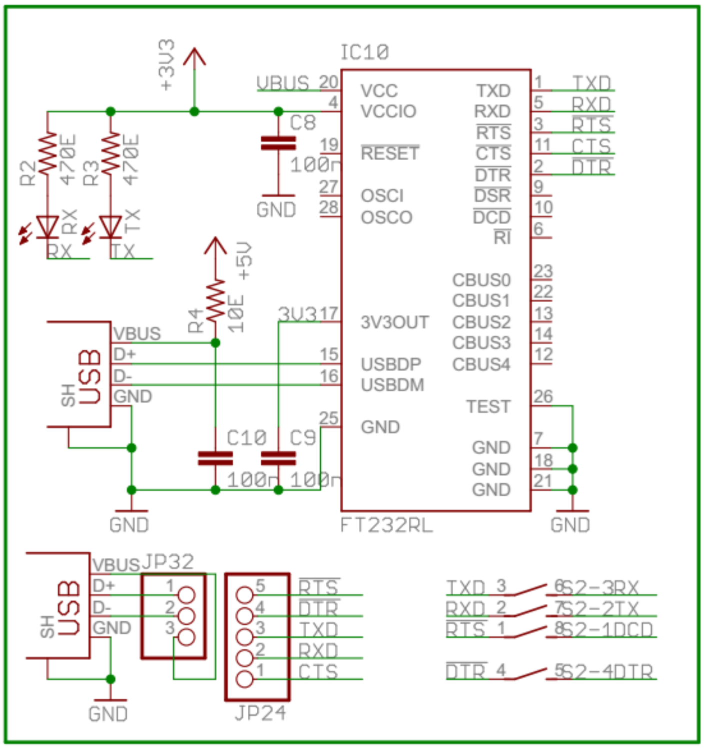
When the high side transmits a '0' (0V) the MOSFET substrate diode conducts pulling the low side down to approx 0.7V, this is also low enough to turn the MOSFET on, further pulling the low side down. When the high side transmits a '1' (5V) the MOSFET source pin is pulled up to 3.3V and the MOSFET is OFF.

Note: This works with I2C and other open collector type gates.



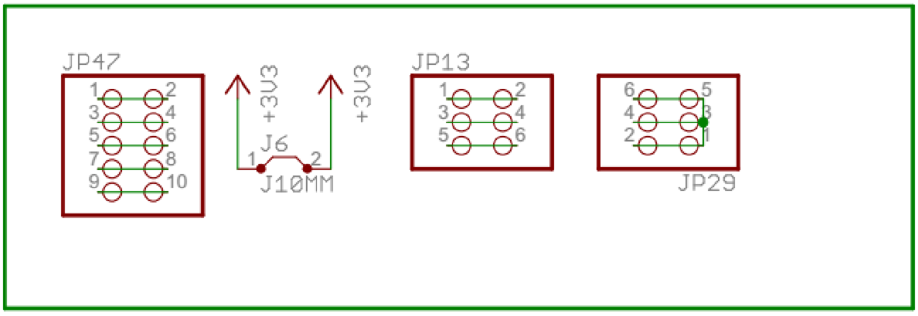
FT232 USB Programming port/converter:

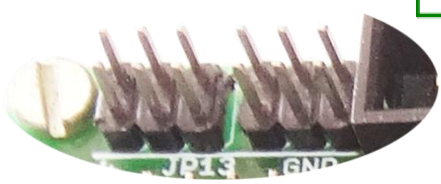
This connection can be used for programming the ARM board or also could be used as a USB to TTL adapter based on the DIP switch state.



Node connector:

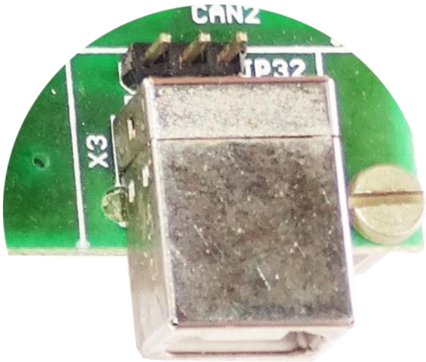
Node connector is an additional on board connection extender or 1 connection IN and 1 connection OUT

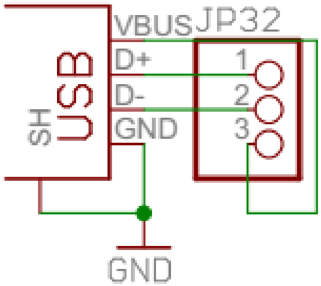




B Type USB Connector:

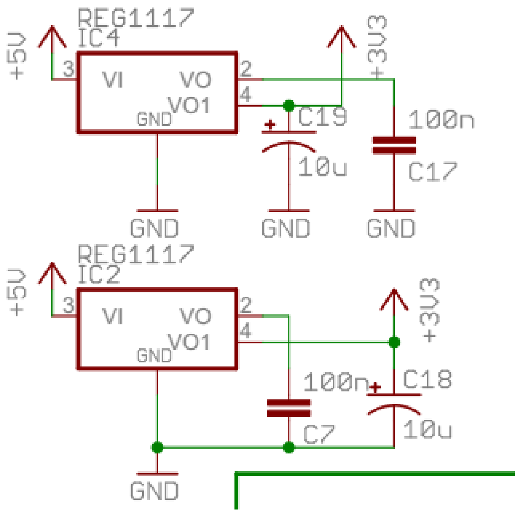
This connector makes use of USB D+ and D- to connect directly to the respected ARM port pins via jumper cables.





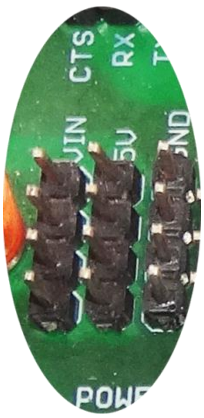
DC 3.3V connectors

These connectors provide on board 3.3V DC connections.



DC 12V 5V connectors:

These connectors provide on board 12V 5V DC connections.



RDL BUS:

12C bus:

One IC that wants to communicate to another must: (Protocol)

1) Wait until it sees no activity on the I2C bus. SDA and SCL are both high. The bus is 'free'.

2) Put a message on the bus that says 'it's mine' - I have STARTED to use the bus. All other ICs then LISTEN to the bus data to see whether they might be the one who will be called up (addressed).

3) Provide on the CLOCK (SCL) wire a clock signal. It will be used by all the ICs as the reference time at which each bit of DATA on the data (SDA) wire will be correct (valid) and can be used. The data on the data wire (SDA) must be valid at the time the clock wire (SCL) switches from 'low' to 'high' voltage.

4) Put out in serial form the unique binary 'address'(name) of the IC that it wants to communicate with.

5) Put a message (one bit) on the bus telling whether it wants to SEND or RECEIVE data from the other chip. (The read/write wire is gone!)

6) Ask the other IC to ACKNOWLEDGE (using one bit) that it recognized its address and is ready to communicate.

7) After the other IC acknowledges all is OK, data can be transferred.

8) The first IC sends or receives as many 8-bit words of data as it wants. After every 8-bit data word the sending IC expects the receiving IC to acknowledge the transfer is going OK.

9) When all the data is finished the first chip must free up the bus and it does that by a special message called 'STOP'. It is just one bit of information transferred by a special 'wiggling' of the SDA/SCL wires of the bus.

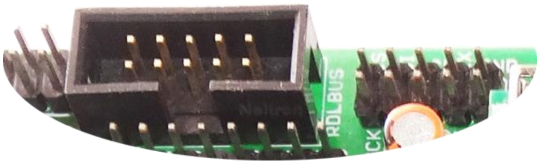
SPI bus:

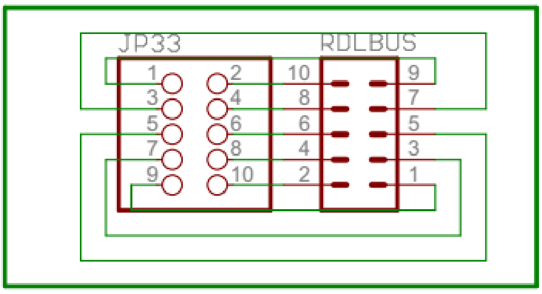
Serial to Peripheral Interface (SPI) is a hardware/firmware communications protocol developed by Motorola and later adopted by others in the industry. Microwire of National Semiconductor is same as SPI. Sometimes SPI is also called a "four wire" serial bus.

The Serial Peripheral Interface or SPI-bus is a simple 4-wire serial communications interface used by many microprocessor/microcontroller peripheral chips that enables the controllers and peripheral devices to communicate each other. Even though it is developed primarily for the communication between host processor and peripherals, a connection of two processors via SPI is just as well possible.

The SPI bus, which operates at full duplex (means, signals carrying data can go in both directions simultaneously), is a synchronous type data link setup with a Master / Slave interface and can support up to 1 megabaud or 10Mbps of speed. Both single-master and multi-master protocols are possible in SPI. But the multi-master bus is rarely used and looks awkward, and is usually limited to a single slave. The SPI Bus is usually used only on the PCB. There are many facts, which prevent us from using it outside the PCB area. The SPI Bus was designed to transfer data between various IC chips, at very high speeds. Due to this high-speed aspect, the bus lines cannot be too long, because their reactance increases too much, and the Bus becomes unusable. However, its possible to use the SPI Bus outside the PCB at low speeds, but this is not quite practical.

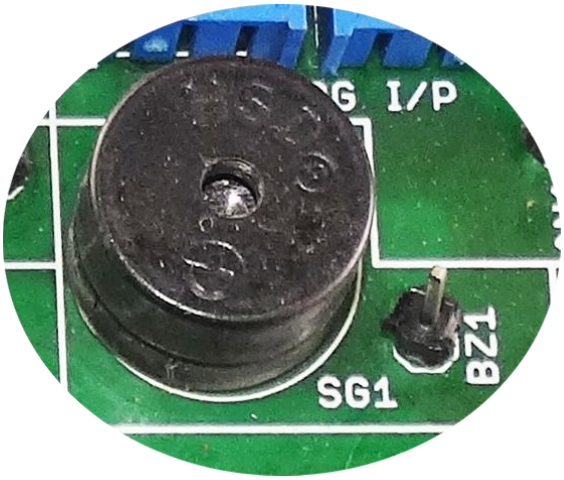
The peripherals can be a Real Time Clocks, converters like ADC and DAC, memory modules like EEPROM and FLASH, sensors like temperature sensors and pressure sensors, or some other devices like signal-mixer, potentiometer, LCD controller, UART, CAN controller, USB controller and amplifier.

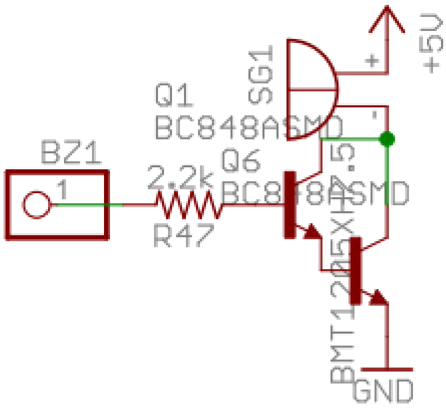




Buzzer:

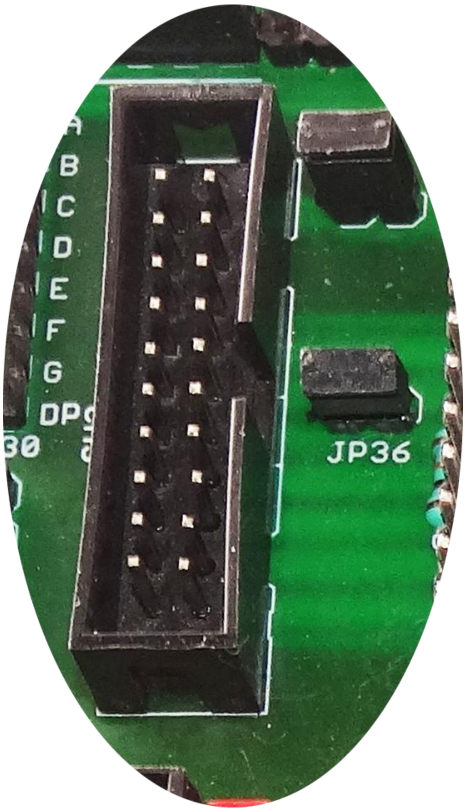
5V continuous tone buzzer, output frequency approx 2300Hz. Diameter 10mm.

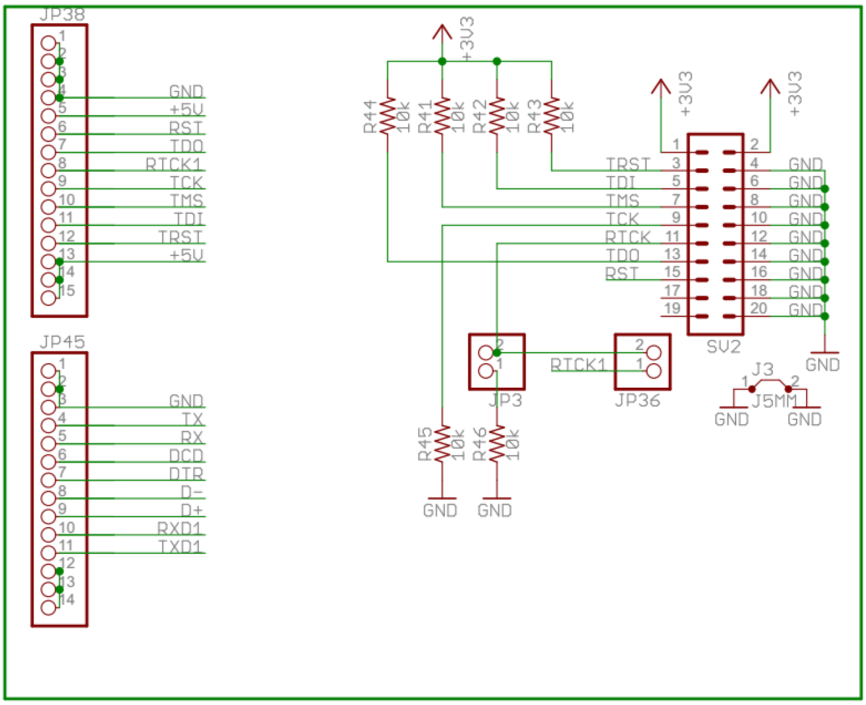




JTAG Debugging with LPC2129

JTAG can be used to program and debug the controller. It can be used to set up break points and step, through the code. The Explore ARM7 (LPC2129), has a ARM 20 pin JTAG connector.



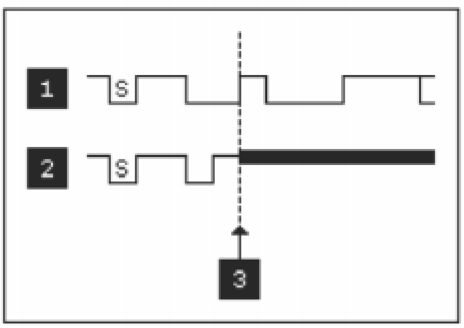


CAN protocol module:

The CAN communication protocol is a CSMA/CD proto-col. The CSMA stands for Carrier Sense Multiple Access. What this means is that every node on the net-work must monitor the bus for a period of no activity before trying to send a message on the bus (Carrier Sense). Also, once this period of no activity occurs, every node on the bus has an equal opportunity to transmit a message (Multiple Access). The CD stands for Collision Detection. If two nodes on the network start transmitting at the same time, the nodes will detect the 'collision' and take the appropriate action. In CAN protocol, a non-destructive bitwise arbitration method is utilized. This means that messages remain intact after arbitration is completed even if collisions are detected. All of this arbitration takes place without corruption or delay of the higher priority message. There are a couple of things that are required to sup-port non-destructive bitwise arbitration. First, logic states need to be defined as dominant or recessive. Second, the transmitting node must monitor the state of the bus to see if the logic state it is trying to send actually appears on the bus. CAN defines a logic bit 0 as a dominant bit and a logic bit 1 as a recessive bit. A dominant bit state will always win arbitration over a recessive bit state, therefore the lower the value in the Message Identifier (the field used in the message arbitration process), the higher the priority of the message. As an example, suppose two nodes are trying to transmit a message at the same time. Each node will monitor the bus to make sure the bit that it is trying to send actually appears on the bus. The lower priority message will at some point try to send a recessive bit and the monitored state on the bus will be a dominant. At that point this node loses arbitration and immediately stops transmitting. The higher priority message will continue until completion and the node that lost arbitration will wait for the next period of no activity on the bus and try to transmit its message again.

WORKING

As stated earlier, CAN is a peer-to-peer network. This means that there is no master that controls when individual nodes have access to read and write data on the CAN bus. When a CAN node is ready to transmit data, it checks to see if the bus is busy and then simply writes a CAN frame onto the network. The CAN frames that are transmitted do not contain addresses of either the transmitting node or any of the intended receiving node(s). Instead, an arbitration ID that is unique throughout the network labels the frame. All nodes on the CAN network receive the CAN frame, and, depending on the arbitration ID of that transmitted frame, each CAN node on the network decides whether to accept the frame. If multiple nodes try to transmit a message onto the CAN bus at the same time, the node with the highest priority (lowest arbitration ID) automatically gets bus access. Lower-priority nodes must wait until the bus becomes available before trying to transmit again. In this way, you can implement CAN networks to ensure deterministic communication among CAN nodes.



1 Device A: ID = 11001000111 (647 hex)
2 Device B: ID = 11011111111 (6FF hex)
3 Device B Loses Arbitration; Device A Wins Arbitration and Proceeds
S = Start Frame Bit

High-Speed CAN Transceiver MCP2551 MCP2551 is the IC used in this board as a CAN transceiver and its features and pin descriptions are given below:

FEATURES

1. Supports 1 Mb/s operation
2. Implements ISO-11898 standard physical layer requirements
3. Suitable for 12V and 24V systems
3. Externally-controlled slope for reduced RFI emissions
4. Detection of ground fault (permanent Dominant) on TXD input
5. Power-on Reset and voltage brown-out protection
6. An unpowered node or brown-out event will not disturb the CAN bus
7. Low current standby operation
8. Protection against damage due to short-circuit conditions (positive or negative battery voltage)
9. Protection against high-voltage transients
10. Automatic thermal shutdown protection
11. Up to 112 nodes can be connected
12. High-noise immunity due to differential bus implementation.

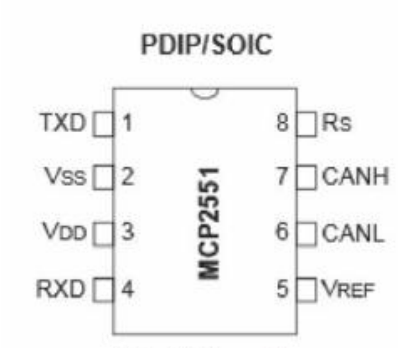


Fig: CAN module

PIN DESCRIPTION

1. TXD :- Transmit Data Input
2. Vss :- Ground
3. Vdd :- Supply Voltage
4. RXD :- Receive Data Output
5. Vref :- Reference Output Voltage
6. CANL :- CAN low-level Voltage I/O
7. CANH :- CAN high-level Voltage I/O
8. Rs :- Slope-control Input