174 Roe Street, PERTH. Western Australia. 6000
Phone: (08) 9428 2188 Fax: (08) 9428 2187
E-mail: kits@altronics.com.au

K 9655

Arduino-Compatible 9-Button Wallplate with LCD

I/O Interfacing is one of those things that almost every project needs. The K9655 provides some of the basics, integrating a 3x3 keypad, 2 relays, a buzzer and 16x2 character LCD into an Australian standard wall plate. Also includes Arduino shield headers for added flexibility.

The Arduino form factor enables a huge range of projects, but finding an enclosure for these projects is often a challenge. The K9655 is a wall mounted (or simply an interface point on another enclosure) Arduino compatible unit with a number of common IO interface options integrated, ready for any project thrown at it.

Theory of Operation:

To drive a 16x2 character LCD requires a minimum of 6 pins to drive, 7 if using the backlight. Driving a 3x3 keypad similarly requires 6 IO capable pins. With the 3 pins required to drive the 2 relays and buzzer, that would leave only 4 of the ATmega328p's 20 digital pins free for other tasks. To enable these functions while still retaining the flexibility of Arduino expansion shields the MCP23S17 IO expander is used to provide 16 additional input and output capable pins. The MCP23S17 is driven by the ATmega328p's SPI bus. Since many shields already make use of the SPI bus, this effectively enables an additional 16 IO for the cost of 1 (4 if not using the SPI bus).

Driving the MCP23S17 involves programming it's registers via SPI commands, setting the states of the individual pins; input/output, pullup/none, active-high/open-drain etc. The pin values of the inputs and outputs are also read/set using these same commands. For a more detailed description of the MCP23S17 operation, either refer to the IC datasheet (available from Microchip) or the library in the example code (available for download on the Altronics website), which implements a number of functions used to controll the MCP IC. A typical SPI bus can only support one master device, and mutiple devices slaved to the

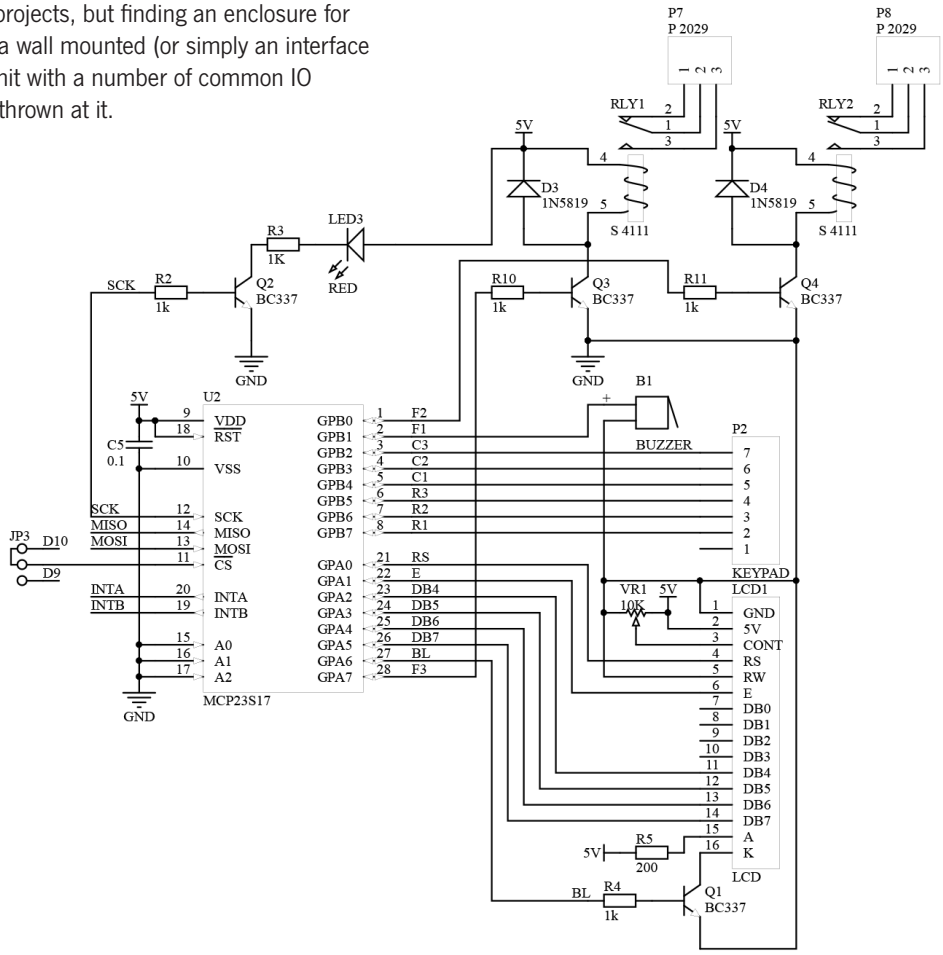


Fig 1. The heart of the unit's functionality is the MCP23S17 IO expander, responsible for providing the driving signals for the LCD, keypad, relays and buzzer. This IC is driven by the ATmega328p's SPI bus, requiring only 4 pins, 3 of which are re-used when driving additional SPI devices.

master unit, each with their own chip select line. In the K9655 the ATmega328p acts as the master, with the MCP23S17 only able to act as a slave unit. The ATmega328p's SPI peripheral has a quirk where it will enter slave mode if its chip select line is set as an input, and is asserted. To avoid this, the hardware chip select (Arduino D10) must be set as an output, regardless of it being used or not. A number of shields make use of this line, so a jumper has been provided to allow an alternative chip select line. However D10 needs to be set as an output in software regardless of it being used or not.

In order to read a matrix keypad, like the one used here, a single row is set as an output and driven low, while the remaining 5 pins are set to input, and held high by internal pullups. If a button is pressed on the row being scanned, the button contacts will cause the input to be pulled low as it gets connected to the low output pin. A read of the column values will now indicate the column of the press, which in combination with knowledge of the row being driven, indicates the unique position of the button being pressed. This procedure is cycled through all 3 rows to perform a full 2D read of the matrix.
If mutiple buttons have been pressed, multiple inputs will report low values. In this situation, for most reads there can potentially be multiple button presses which will produce the same values. In order to account for this, it is best to simply ignore reads where multiple presses are detected.

PCB Construction:

A jumper has been provided to control the chip select line for the MCP23S17. This jumper should be soldered first since it becomes difficult to access once the components either side have been soldered in. Pin 10 is the hardware chip select line from the ATmega328P, and should be used wherever possible. However it is commonly used on shields (such as the Ethernet shield), so Pin 9 has been provided as an alternative. Decide what configuration the jumper should be in and close it using solder. The code pre-programmed onto the ATmega328p assumes that pin9 is used.

The first components to be fitted is the only SMD component: the MCP23S17 IC. Thankfully the IC has relatively well spaced legs, so shouldn't prove too dificult to solder. Position the IC on the footprint and hold in place with either a clothes peg or some Bluetac. If needed, the IC can be nudged slightly untill it is centered on the footprint. Once positioned adequately, solder two legs, one on each opposing corner of the IC. If the IC needs further adjustment in its positioning, the solder on one of the pads can be re-melted, and the IC nudged into position, with the remaining leg holding the IC to the board. Once the IC is positioned adequately solder the remaining legs to the PCB. A fine-tipped soldering iron is preferable for SMD soldering work, however due to its size this IC should prove rather forgiving.

In order to fit all components in the available space, the crystal for the ATmega has been positioned on the bottom of the PCB, sitting between the PCB and the wall plate. In its position it covers a number of solder pads which have to be populated first and then cut as flush as possible. To begin construction, fit the 1M resistor (R6), 2 22pF capacitors (C6 & 7) and IC socket and solder to the PCB. For the IC socket, it is advisable to solder a pin in each corner initally (in the same way as the MCP23S17), and ensure the socket is seated flat before soldering the remaining pins. Trim the component legs around the crystal's silk screen marking as close to the PCB as possible.

Cut a small section of the provided double-sided tape and place it so it covers the silk screen outline of the crystal. This will prevent any accidental shorts caused by the metal body of the crystal. The crystal can now be fitted to the bottom side of the PCB, poking holes through the tape in the process (alternatively a sewing needle can be used to form the holes, or the crystal legs cut on a slight angle to give them a sharp point).

The next component to fit is the header for the LCD. This header has a very fine pitch pins, which will make it difficult to solder with only a wide chisel tip soldering iron. It is

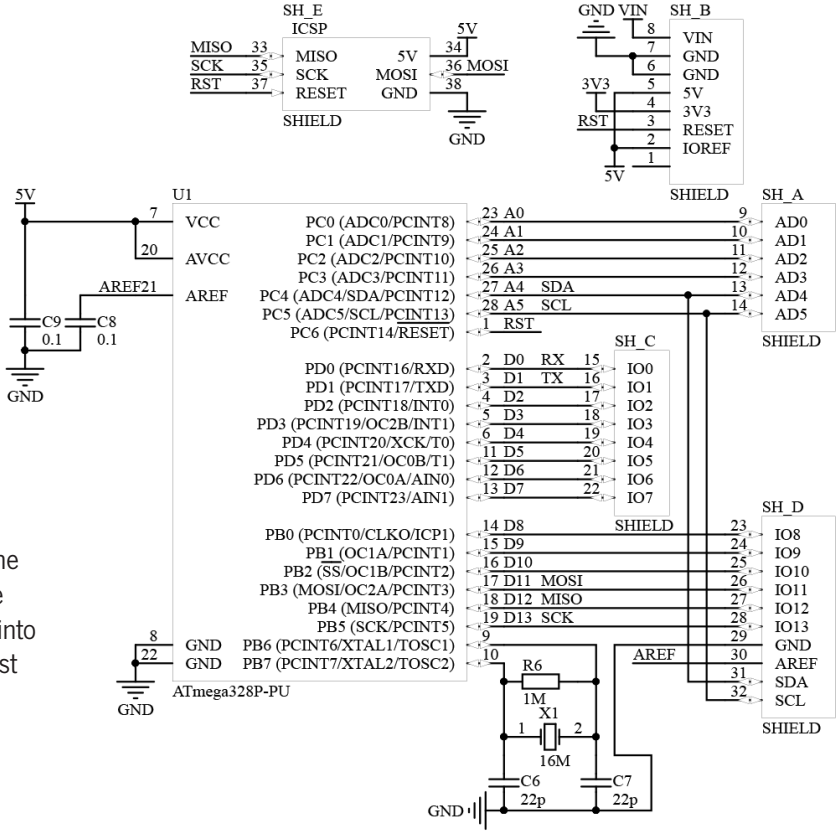


Fig 2. Since the MCP23S17 does all the heavy lifting, there are very few connections into the ATmega328p. Most pins are simply wired directly to the shield expansion headers.

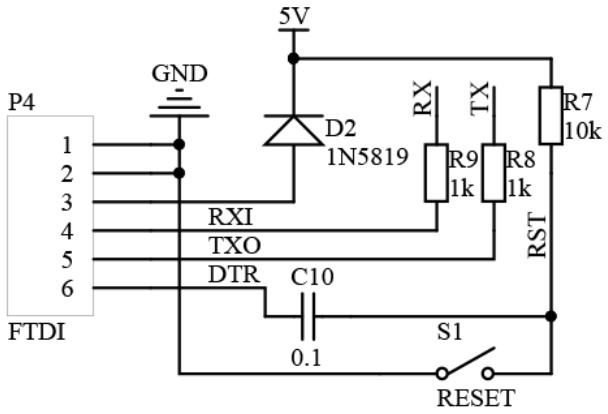


Fig 3. To minimise the components on the PCB, the unit is programmed with a standard FTDI breakout unit, plugged into the on-board header.

highly recommended to use the finest point tip and the thinnest gauge solder available. The header is non-polar, and can be fitted in any orientation.

The remaining components can now be fitted and soldered as normal, starting with the shortest and those unable to hold themselves in place while being soldered. Firstly, fit the 7-pin right-angle header for the membrane keypad. This header should sit facing over the MCP23S17. Now the 3.5mm terminal blocks for the relays and power input can be fitted, followed by the pin and socket headers for the FTDI connection and UNO shield headers, buzzer and relays. To fit the 7805 regulator the legs are bent so the regulator can sit flush against the PCB. Hold the regulator in its position, and mark the legs where they need to be bent. Bend the legs, fit the regulator with the supplied Silpad underneath, and using the polycarbonate M3x6mm bolt and polycarbonate hex nut, bolt the regulator to the PCB.

While the K9655 won't fit inside an Australian standard wall box due to the width of the Arduino header spacing, it will fit neatly within a Mounting Block, such as the P8054. In order for it to fit, a number of the resistors and diode need to be bent down to sit flush against the PCB. These are; D2, R7, R10 (Bent up) & R11. Fit the remaining passive components now along with the diodes, LEDs and transistors, according to Fig.8 (on cover sheet), bending them flat against the PCB where necessary. At this point, the only remaining components should be the reset button, contrast potentiometer and 3.3V regulator. Once theses have been fitted the PCB assembly is almost complete.
The back of the PCB rests against the LCD screen, and is used to hold it in position. To prevent shorts from contact with the metal housing on the LCD, the component legs in the area are trimmed flush to the PCB and double sided tape is placed over the top to insulate the joints. The contact area is from just above the LCD header on the PCB, to just below the bottom of the T cutout on the board for the LCD and keypad leads. Trim the legs of all components between these two lines (visible on the top of the PCB) and using the supplied double-sided tape, run 2 lengths across the board to cover the exposed solder joints.

The PCB assembly is now complete.

Final Assembly:

To assemble the wall plate, firstly fit one of the provided covers. This will provide a guide border for sticking down the membrane keypad. Peel the protective layer from the rear of the keypad, and attach it to the wall plate, with the lead heading through the line cutout on the top of the wall plate. Next is to temporiarially mount the LCD using the adhesive on the membrane keypad. The solder joint connecting the LCD cable to the LCD itself should be at the top of the wall plate, the same side as the membrane keypad cable. Remove the protective film from the LCD screen, and stick it down into the cutout. The adhesive on the keypad will hold the LCD in place, with the pressure from the PCB ensure it can't come loose. Fold the LCD cable up so it can be run alongside the the keypad cable.

Using 4 M3x10mm countersunk bolt, fit them through the 4 countersunk holes on the wall plate, and fasten them tight with 4 polycarbonate nuts. The PCB will be held in place by these bolts, with the first nuts acting as standoffs to keep the PCB level. Feed the LCD and keypad cable through the T cutout on the PCB, and seat the PCB on the 4 bolts. There will be a small length of the bolt left on which another set of polycarbonate nuts is fitted to hold the PCB in place. These nuts don't need to be tightened yet, and can be left off entirely untill the unit has been tested and confirmed working.

Both the LCD and keypad cables can now be fitted to their respective connectors. Both should attach with the cable straight (keypad is offset by a few milimeter, but this won't affect attachment), with no bends or twists.

The unit assembly is now complete, and ready for testing.

Testing:

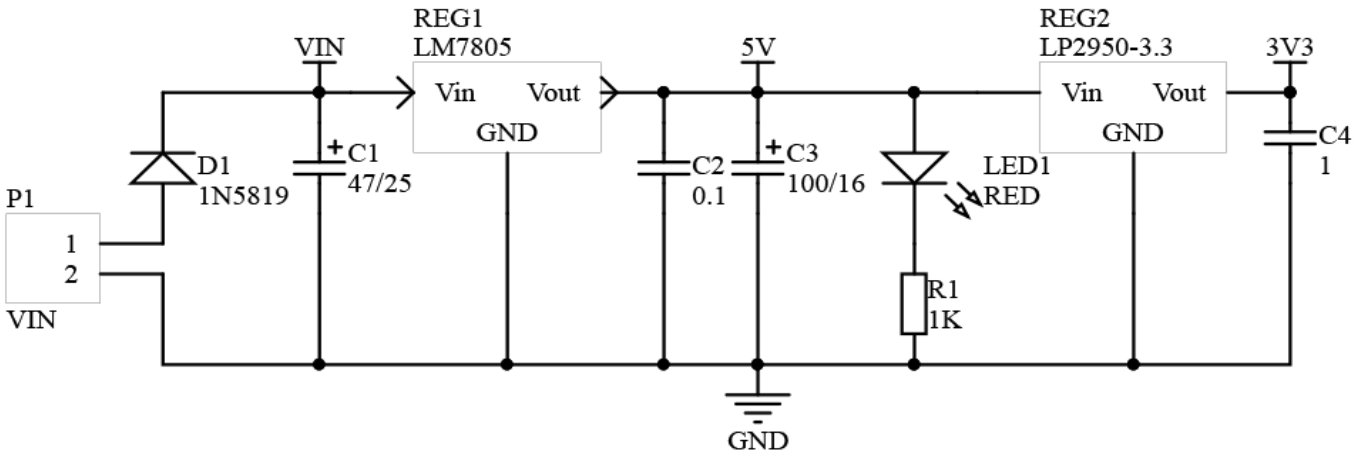
The unit comes pre-programmed with a code example making use of all the included functions of the unit. The only setup which is needed is to adjust the contrast potentiometer. The code uses D9 as the chip select line for the MCP23S17, so if D10 has been selected by the previously soldered jumper the code will need to be downloaded and modified to use the new pin. This is done by changing the definition on line 4.

To apply power, a 9-15V source can be applied to the Vin pin on the Arduino header, or to the screw terminals on the left of the PCB. Alternatively, a regulated 5V source can be applied between the 5V and GND pins on the Arduino header. With power applied to the unit the backlight on the LCD should be lit by the running code. If it does not, power down the unit and double check component placement and soldering. If it does, adjust the potentiometer untill the text is clearly visible. The poteniometer should be rotated approximately one quarter turn clockwise, but this will depend on specific component tollerances, and possibly lighting conditions.

From here, the buttons on the keypad should be shown on the screen, and some will cause the relays to toggle and buzzer to sound.

If no activity is visible, double check that one option on the MCP23S17 chip select jumper has been soldered. Uploading the Blinky sketch from the Arduino IDE should cause the D13 LED (to the left of the contrast potentiometer) indicating that the ATmega328p is working correctly. From here, check the soldering around the MCP23S17, specifically

Fig 4. A reverse polarity protection diode is provided for the screw terminals, along with solid 5V and 3.3V rails. NOTE: Reverse polarity protection is not provided for Vin, 5V and 3.3V supplied to the Arduino header.



for shorts on the pins of the MCP23S17 and LCD header.

To program the K9655, select the Arduino Duemilanove from the IDE. The unit will need an external FTDI, such as the Z6212 to handle the USB to serial conversion. The header is positioned such that the unit can still be reprogrammed in a Mounting Block if necessary.

Important Note:
Please note that we can offer a warranty only on the components supplied with this kit. Because we are unable to guarantee your labour, there is no warranty on either partially or fully built kits. We are able to offer a repair service, but once construction has commenced, this service is chargeable.

Dear Kit Constructor,
At Altronics we take great pride in the quality and presentation of our kits. If you find any deficiency in this kit or have any constructive comments whatsoever, please write to us.

kits@altronics.com.au

Overlay Diagram