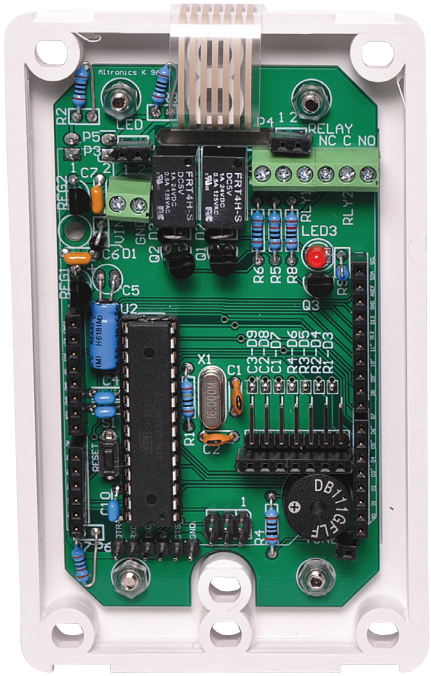
ALTRONICS
174 Roe Street, PERTH.
Western Australia. 6000
Phone: (08) 9428 2188
Fax: (08) 9428 2187
E-mail: kits@altronics.com.au
Dealers Throughout Australia

K 9650

Arduino-Compatible 12-Button Wallplate

I/O Interfacing is one of those things that almost every project needs. The K9650 provides some of the basics, integrating a 4x3 keypad, 2 relays and a piezo transducer into an Australian standard wall plate. It also includes an Arduino compatible ATmega328P and Arduino shield headers for added flexibility.



The main feature of the K9650 is the wallplate sized membrane keypad. Electrically, the keypad is an array of 12 switches arranged into a 3-column, 4-row grid. Reading keypads like this requires multiplexing to individually read all switches along a single line at once, else a single switch could be read as multiple switches. To handle this multiplexing in software, the Arduino "Keypad" library (Search "Keypad "from the Arduino IDE's Library Manager) can be used.

Solder jumpers are included (NO type) to reduce the number of wires needed to attach the keypad, while the breakout header can be used to remedy pin-conflicts with any shield used. C1-3 indicates the column connected, R1-4 indicates row and D3-9 indicates the digital pin the jumper connects to. '1' on the keypad is on the R1-C1 position.

Feedback options include 2 1A relays, driven via BC547 transistors, 2 high brightness LEDs fitted so they shine through the membrane keypad and a piezo transducer. The Arduino standard D13 LED has also been included as a diagnostic tool (LED3). Driving the LEDs and relays can be managed via the standard digitalWrite command thanks to the transistors buffering the relays. However, being a transducer the piezo requires an oscillating waveform. Arduino's built in "Tone" library can be used to generate square waves at a specified frequency, while more advanced DDS code examples exist, which can generate higher precision sine waves (an external RC filter would likely be needed to smooth out the switching waveform).

Constructions

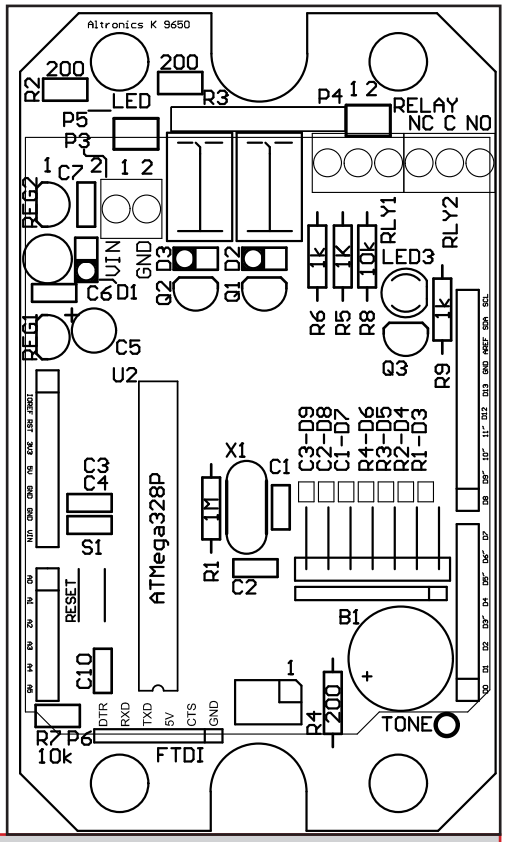
To begin construction, close the required solder jumpers for the membrane keypad, then fit the resistors and diodes: R1;1MΩ, R2-4;200Ω, R5,6,9;1kΩ, R7,8;10kΩ. Note than R2,3,7&9 are fitted vertically, and bent over (Up for R2,3, down for R7, R9 is not bend, but should be fitted to minimise height) as they stand too tall when a wall box is fitted. D1-3 are all fitted vertically, with the cathode stripe matching the silkscreen fill on the board.

Fit the capacitors next, making sure to pay attention to the polarity and mounting position of C5; The stripe indicating the negative lead should point into the centre of the board. Leave enough height to bend C5 down against the board to avoid height problems when a shield is fitted. The remaining capacitors are ceramic types, and are not polarity sensitive: C1,2;22pF, C3,4,10;100nF, C5;10µF/50V, C6,7;2.2µF.

The crystal (X1) and reset button (S1) can also be fitted now. They are not polarity sensitive.

Next, fit the transistors and voltage regulators, ensure that they aren't mixed. They should be fitted as close to the board as possible to ensure there aren't any clearance problem to a fitted shield. Q1-3 are BC547, fitted with the flat marking side matching the flat on the silkscreen. REG1 is the 5V regulator and is indicated by the "KY5050" marking on the package. Reg2 is the 3.3V regulator and similarly is marked "KY5033". They are both fitted with the flat side pointing out from the board. LED3 can also be fitted now (Red, not the water-clear LEDs), making sure the flat on the LED matches the flat on the silkscreen.

At this point, the only remaining top-side components should be the relays, headers, IC socket, and screw terminals. Fit the screw terminals, IC socket (notch to bottom of board) and headers now. The screw terminals are best fitted with the cable entry to the bottom of the board. The 6, 8 (x2) and



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

10pin sockets are to be fitted to the Arduino shield footprint. The 7-pin male header is fitted to the FTDI interface point (P6), with the right angle male header fitted to the keypad attach point (pins pointing to top of board), immediately right of C2. Cut the supplied length of socket pins into lots of 7,2,2&1 and fit these to the keypad breakout (7), relay interface (top right, 2), LED interface (top left, 2) and piezo (bottom right, 1).

Fit the relays now to complete assembly of the top of the board.

Check the colour of the clear LEDs with a multimeter's diode test function (Red and Green), then fit them to the positions immediately left of the LED interface on the underside of the board. The flat of the LED should be pointed towards the bottom of the board. It is important that these LEDs be fitted so the body of the LED is hard up against the board.

It is a good idea to test the board at this stage, before mounting it to the wall-plate. The IC comes pre-programmed with the Arduino boot-loader and our example sketch. As a heartbeat indicator, it blinks the D13 LED, so no additional wiring is necessary.

Mount the IC into the socket and connect 6-24V into the Vin-GND screw terminal. It is best to use a current limited supply (or 9V battery) if available with a very low current limit set. At Idle the circuit should draw a few milliamp. If the LED doesn't flash at 0.5Hz (1s on, 1s off), double check the components have been fitted in the correct place, and the solder joints make proper contact between pad and leg.

Final Assembly

The LEDs on the bottom of the board will poke through the wall-plate when mounted, potentially causing the corner of the membrane keypad to lift. To remedy this, the board can be mounted to the wall-plate and the top of the LED cut flush with a sharp Stanley blade, or washers can be used to stand the board off slightly. If using washers (not supplied), make sure these are plastic types so they wont conduct if touching the pads on the board.

Remove the adhesive protective layer from the back of the keypad and fit it to the supplied wall-plate, with the cable poking through the slot in the top of the plate. If not already done, mount the board to the wall-plate using countersunk M3 screws and nuts. Finally, plug the keypad into the right angle header (The cable should not have any bends in it, although it is offset), and the unit is ready to go.

Software

The micro controller comes pre-programmed with an example program. The source can be downloaded from the Altronics website, from the K9650 product page. If programming via FTDI, select "Arduino Duemilanove or Diecimila" as the board, and the FTDI's COM port as the port.

The program implements a gate access controller: Fitting a wire link and resetting the unit allows loading of a code, up to 50 digits, which is stored in EEPROM for subsequent use. Upon entering the correct code (within the allowed time) the unit will indicate using the LED, piezo and will trigger the relay for 3.6s, allowing an attached door-stike time to let someone through.

To use the program code supplied, wire up the green LED to D10, red LED to D11, piezo to D12 and one of the relays to D15 (A1).

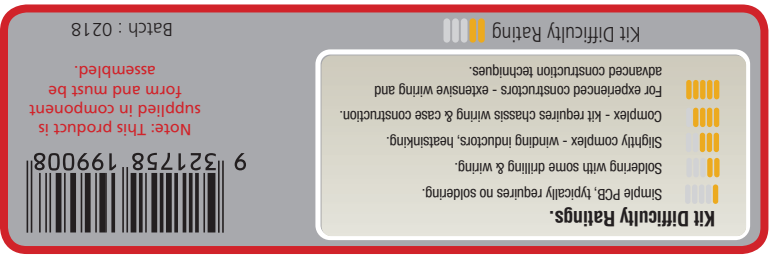
To enter a code, fit a link between A0 (D14) and GND and apply power to the unit (or press reset (S1) if power was applied before fitting link). This will put the unit into code programming mode: Enter the desired code, one digit at a time. Entering '*' will reset the code entry process while entering '#' completes the code entry and resumes normal operation.

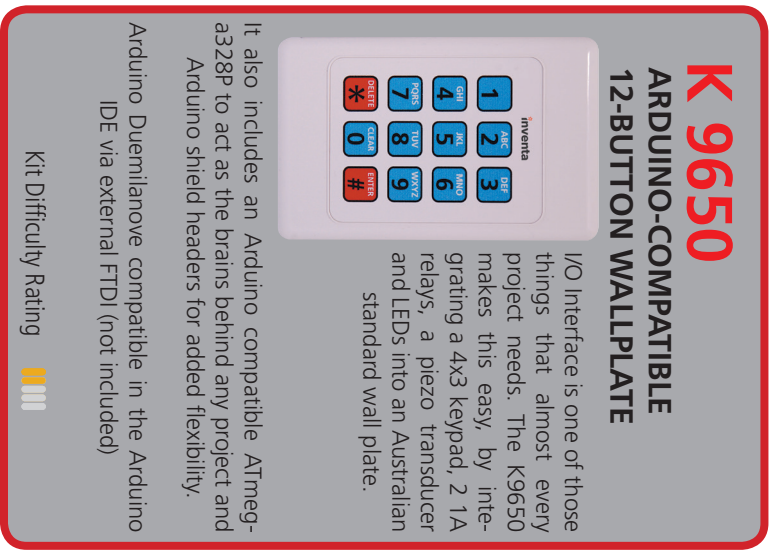
After '#' is pressed, the code will be stored in EEPROM and the link can be removed. The unit will automatically resume normal operation.

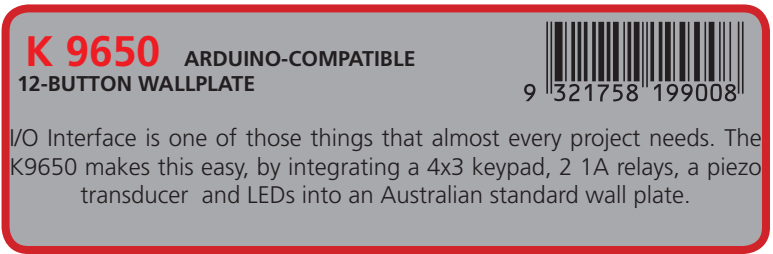
Attempting to enter the code will produce no feedback while entering is in progress for security's sake. A time-out exists as a means of reset, such that each new digit must be entered within 1s of the last, or the unit will time-out the code entry and reset it, indicating the time-out with the red LED and piezo.

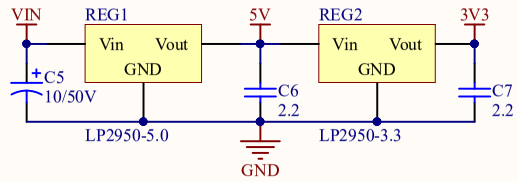
A successful code is indicated by an ascending tone on the piezo, the green LED will flash, and the relay will trigger for 3.6s.

While this example shows one of the many potential used for the kit, it should not be relied on to protect anything of value: There's nothing stopping someone from simply unscrewing the unit and wiring the relay pin into 5V. Fitting an anti tamper switch, which would lock down power to the unit upstream (possibly using a latching relay triggered by the uC) could be used to make a relatively safe and secure unit, but we'll leave that to the constructor to experiment with.

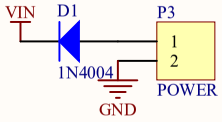




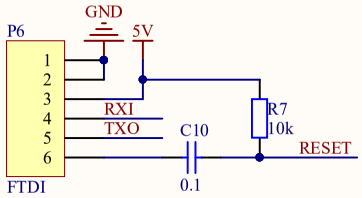




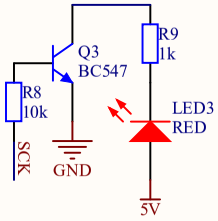
Input polarity protection and power supply



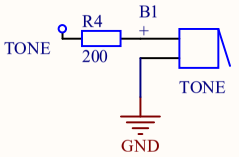
FTDI



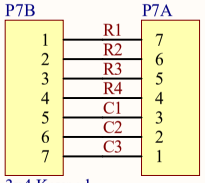
D13 LED

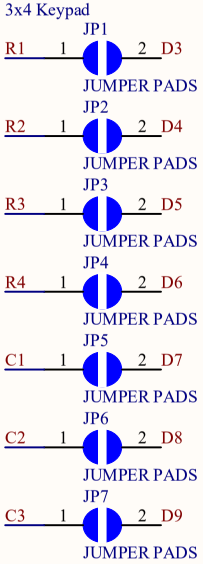


Piezo Interface

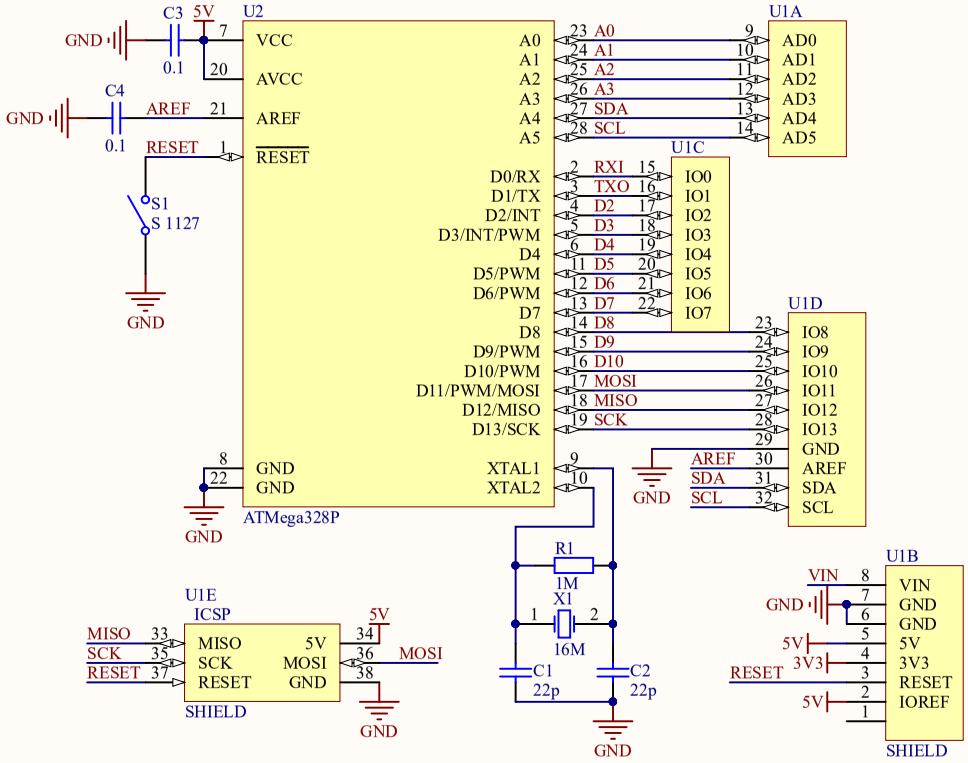


Keypad Interface

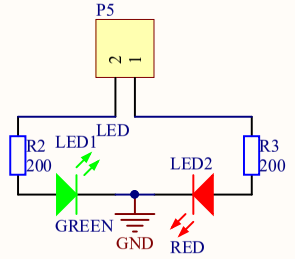




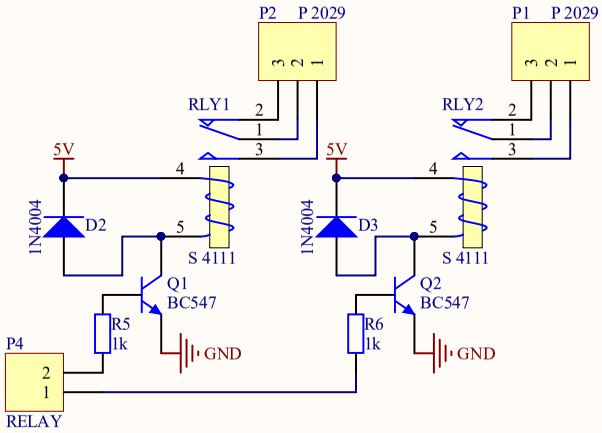
ATmega328P with Arduino shield headers

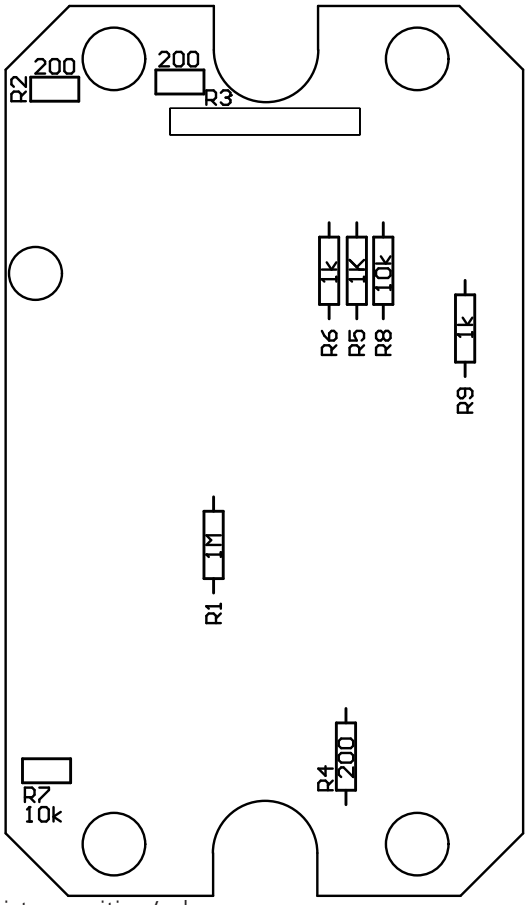


LED Interface



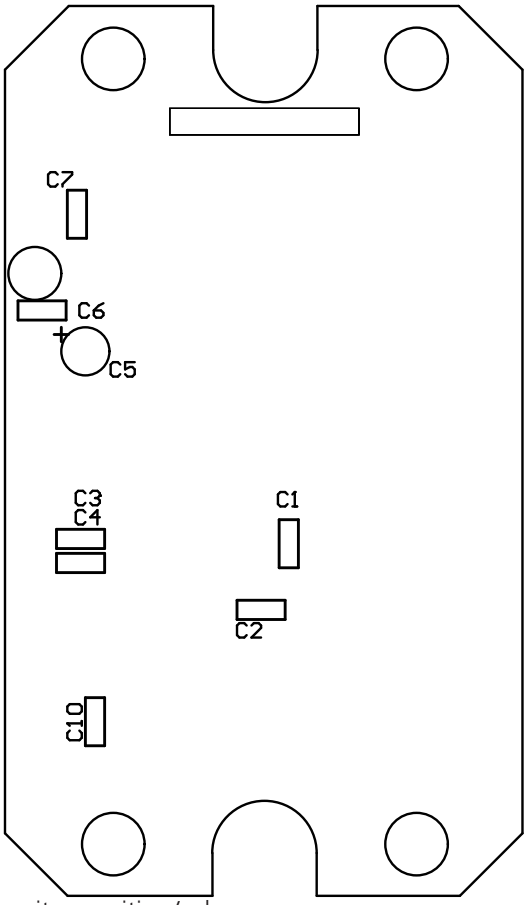
Relay Interface





Resistor position/values

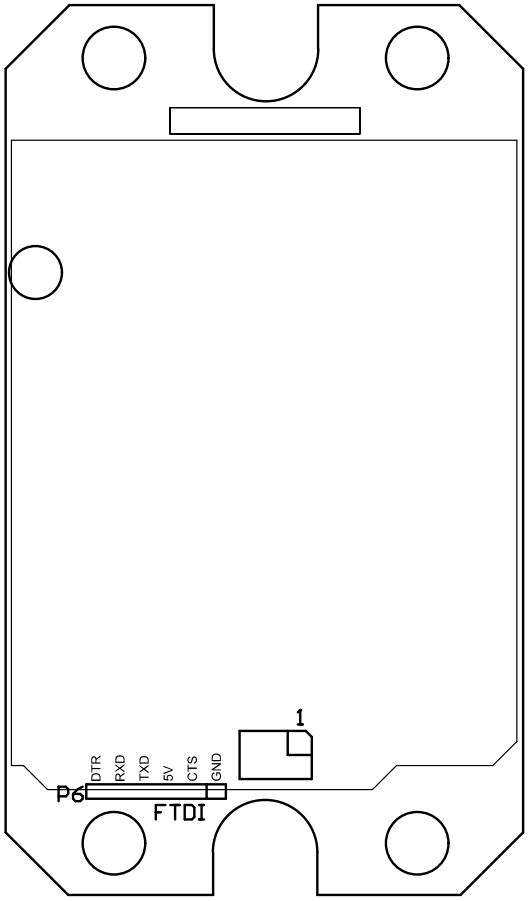
R1 - 1MΩ
R2,4 - 200Ω
R5,6,9 - 1kΩ
R7,8 - 10kΩ



Capacitor position/values

C1,2 - 22pF
C3,4,10 - 100nF
C5 - 10µF/50V
C6,7 - 2.2µF

Male header positions



Female header positions