Micromega Corporation

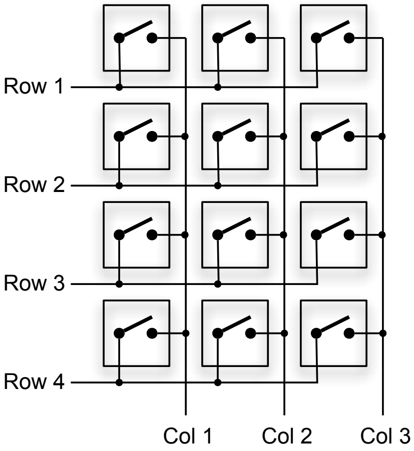
Code Example uM-FPU64

Interfacing Keypad Switches

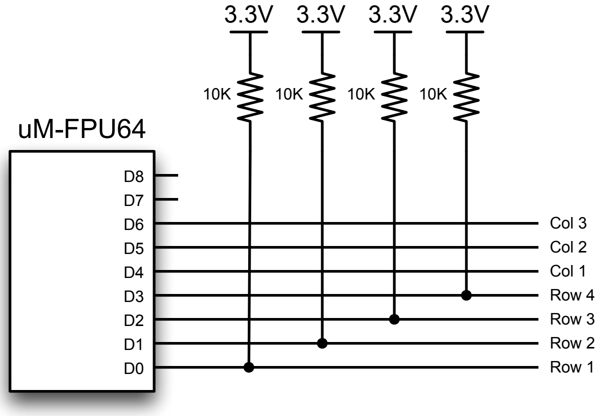
Introduction

This code example provides FPU functions for reading a matrix of switches. When a switch is pressed, the corresponding bit in the 32-bit integer value is set. Any configuration of rows and columns to a maximum of 32 switches can be handled. A 3x4 membrane switch is used in this example. A bit pattern is output sequentially to the columns and the row input values are read to determine if a key is pressed.

3x4 Membrane Switch



Connecting to the FPU



FPU Functions

Arduino test file: *test_keypad.ino*
FPU functions: *keypad.fp4*

```
getkey() long
```

Calls `waitkey` to get the next key pressed, and then uses the `ltable` and `llookup` functions to map the key code to the following values:

0-9	0 to 9 keys
10	* key
11	# key

The `ltable` and `llookup` functions can easily be changed to map other keypad layouts. The key code is ignored if multiple keys are pressed simultaneously.

```
waitkey() long
```

Calls readkey and waits for the first non-zero key code that's different from the last key code, so that holding down a key will not return multiple values.

```
readkey() long
```

Scans the switch array and returns the key code. A bit is set in the return value for each switch that is currently pressed. If no keys are pressed, zero is returned.

uM-FPU64 features shown in Code Example

- parallel input/output to digital pins
- table lookups

Compiler Functions and Procedures	uM-FPU64 Instructions
`digio(WRITE_BITP,…)`	`DIGIO`
`digio(READ_BITP,…)`	`DIGIO`
`ltable(…)`	`TABLE`
`llookup(…)`	`LTABLE`

Further Information

See the Micromega website (http://www.micromegacorp.com) for additional information regarding the uM-FPU64 floating point coprocessor, including:

uM-FPU64 Datasheet
uM-FPU64 Instruction Set
uM-FPU64 IDE User Manual
uM-FPU64 IDE Compiler Manual