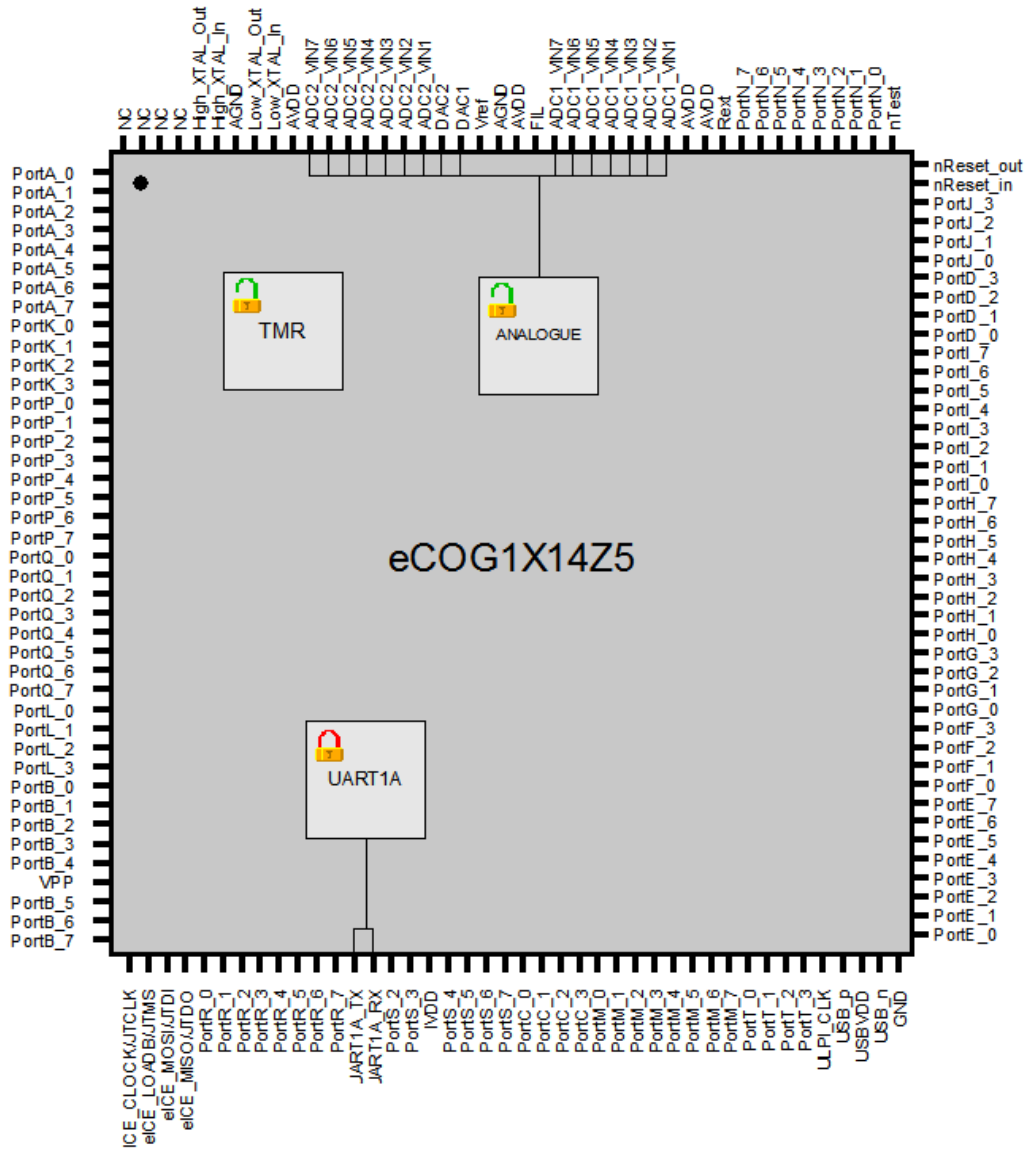




AN063 – Reading a Keypad with the ADC

Version 1.0

This application note describes how to read a 16 key simple keypad using the internal ADC.



Confidential and Proprietary Information

©Cyan Technology Ltd, 2008

This document contains confidential and proprietary information of Cyan Technology Ltd and is protected by copyright laws. Its receipt or possession does not convey any rights to reproduce, manufacture, use or sell anything based on information contained within this document.

Cyan Technology™, the Cyan Technology logo and Max-eICE™ are trademarks of Cyan Holdings Ltd. CyanIDE® and eCOG® are registered trademarks of Cyan Holdings Ltd. Cyan Technology Ltd recognises other brand and product names as trademarks or registered trademarks of their respective holders.

Any product described in this document is subject to continuous developments and improvements. All particulars of the product and its use contained in this document are given by Cyan Technology Ltd in good faith. However, all warranties implied or expressed, including but not limited to implied warranties of merchantability, or fitness for purpose, are excluded.

This document is intended only to assist the reader in the use of the product. Cyan Technology Ltd shall not be liable for any loss or damage arising from the use of any information in this guide, any error or omission in such information, or any incorrect use of the product.

This product is not designed or intended to be used for on-line control of aircraft, aircraft navigation or communications systems or in air traffic control applications or in the design, construction, operation or maintenance of any nuclear facility, or for any medical use related to either life support equipment or any other life-critical application. Cyan Technology Ltd specifically disclaims any express or implied warranty of fitness for any or all of such uses. Ask your sales representative for details.



Revision History

Version	Date	Notes
V1.0	20/08/2007	First release.

Contents

1	Introduction	5
2	Glossary	5
3	Hardware	6
3.1	ADC and Keypad.....	6
3.2	Key Recognition	7
4	Software	8
5	Example Application	8

1 Introduction

This application note describes how to use a single internal ADC channel on the eCOG1X to read a simple 16 key keypad. The purpose is to decode the keys using only one ADC port pin. This has the advantage of reducing the number of pins used when compared to a traditional matrix keypad.

The example code is available as a CyanIDE project and runs on the eCOG1X development board.

2 Glossary

A table of abbreviations used in this document.

ADC	Analogue to Digital Converter
eCOG1	Cyan Technology target micro controller
UART	Universal Asynchronous Receiver/Transmitter

3 Hardware

3.1 ADC and Keypad

The eCOG1X contains two 12-bit ADCs with a maximum of 14 input channels. The eCOG1X ADCs allow the user to select single-ended or differential input mode, with a selectable resolution of 6, 8, 10 or 12bits.

For this example, an analogue keyboard is connected to one of the eCOG1X analogue inputs.

When a key is pressed, the input voltage detected by the ADC changes according to the voltage divider activated by the keypad. The conversion result from the ADC is used to determine which key is pressed.

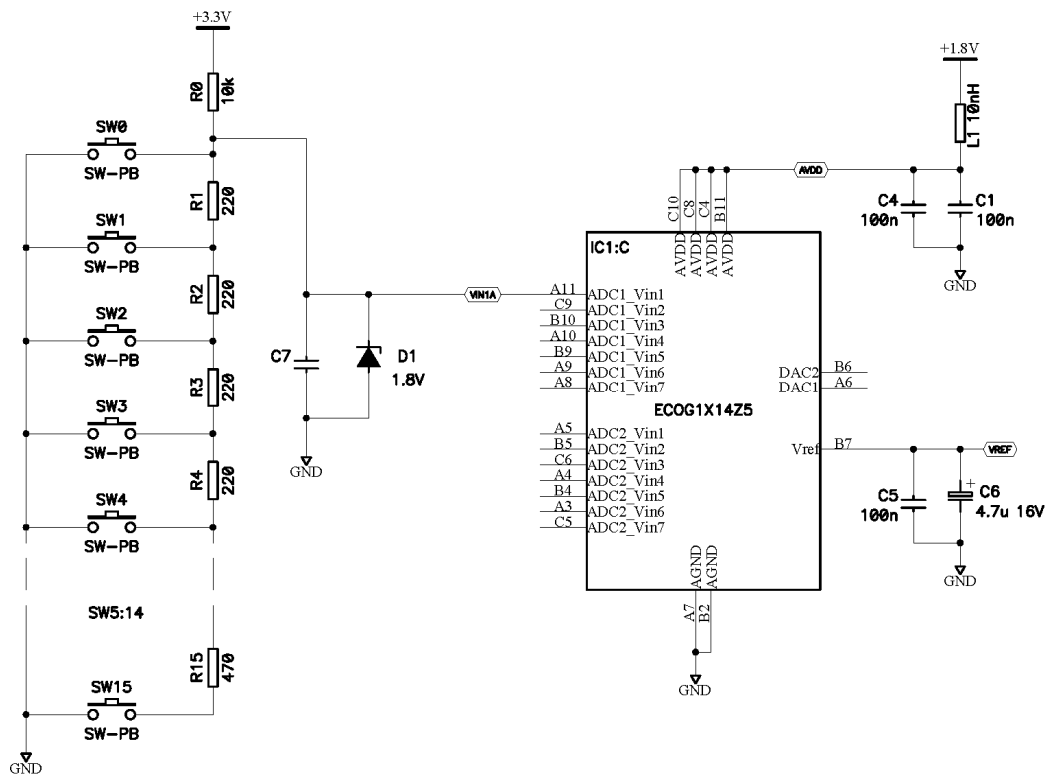


Figure 1. Keypad circuit diagram

Notes:

- R0 is a pull-up resistor by which the input voltage V_{in} is maintained at V_{dd} when no key is pressed. It forms the upper part of a potential divider with $R1..R_n$ when key n is pressed.
- A Zener diode is required to make sure the input signal to the eCOG1X ADC does not exceed the maximum analogue input voltage.

The input voltage associated with each key n is given by:

$$V_{in} = (V_{dd} - V_{ss}) \times \sum(R1..R_n) / (R0 + \sum(R1..R_n))$$

If more than one key is pressed at the same time, the key detected is the closest key to the ADC input in the chain. This means that the key recognition is managed by priority.

3.2 Key Recognition

The number of keys that can be detected depends on the tolerance of the resistors used and the noise from the circuit. The application software uses a voltage range corresponding to each key when converting an analogue reading to a key number.

The resistor values and the equivalent voltages for each key are listed in the following table.

When no key is pressed, the keypad voltage is pulled up to 3.3V by R0. The ADC input voltage is limited by the Zener diode.

The eCOG1X ADC input range is from 0V to Vref (1.22V), which means that an input voltage greater than 1.22V gives a conversion result of 4096. When no key is pressed, the ADC input value is 4096.

The two columns Min and Max show the range of possible ADC input values corresponding to each key. These define the threshold values for the application software, used to decide which key is pressed, allowing the widest possible range for each key allowing for noise and tolerance errors on the keypad resistors.

Vdd = 3.3 V

Vref = 1.22 V

R0 = 10000 Ω

Key	Resistor Ω	Voltage	ADC result	Min	Max
0	0	0.0	0	0	119
1	220	0.071	238	120	353
2	220	0.139	467	354	576
3	220	0.204	686	577	814
4	270	0.281	943	815	1065
5	270	0.354	1187	1066	1304
6	270	0.423	1420	1305	1531
7	270	0.489	1642	1532	1771
8	330	0.566	1900	1772	2022
9	330	0.639	2144	2023	2281
A	390	0.720	2417	2282	2545
B	390	0.796	2673	2546	2794
C	390	0.868	2915	2795	3029
D	390	0.936	3143	3030	3272
E	470	1.013	3401	3273	3522
F	470	1.085	3644	3523	3870
None	∞	1.22	4096	3871	4096

Table 1. Keypad voltages and ADC conversion values

4 Software

In the program main loop, the eCOG1X checks the value of the ADC input and converts it to a key number. The key number is compared with the last recognised key number. If it is the same, then no output is given. If it is different (including the case where no key is pressed), the ADC input is read again after a debounce time (50ms) to allow the input to settle. The reading taken after the debounce time is converted to a key number and output via the serial port.

The ADC is a successive approximation type, with an input sample/hold capacitance of approximately 30pF. If the analogue input source resistance is too high (typically more than 150kΩ), then either the ADC sampling clock may need to be slowed down or the output of the resistive divider must be buffered. An initial estimate for the sampling time is 3x the RC time constant for the source impedance and the sample/hold capacitor.

The threshold values used in the software to determine which key is pressed are shown in the previous table, as calculated from the input voltages and the nominal resistor values. If the ADC result value is greater than 3871, the routine assumes that no key is pressed.

5 Example Application

The example project for this application note demonstrates the detection of 16 keys from the circuit above.

The example displays keys pressed via the serial port at 9600 baud. When a button is pressed by the user, the ADC input value and the corresponding key number are output. The serial port settings are 8 data bits, 1 stop bit and no handshaking.