

# CMOS-Camera based Motion Detection



Wolf Paulus  
([wolf@paulus.com](mailto:wolf@paulus.com))

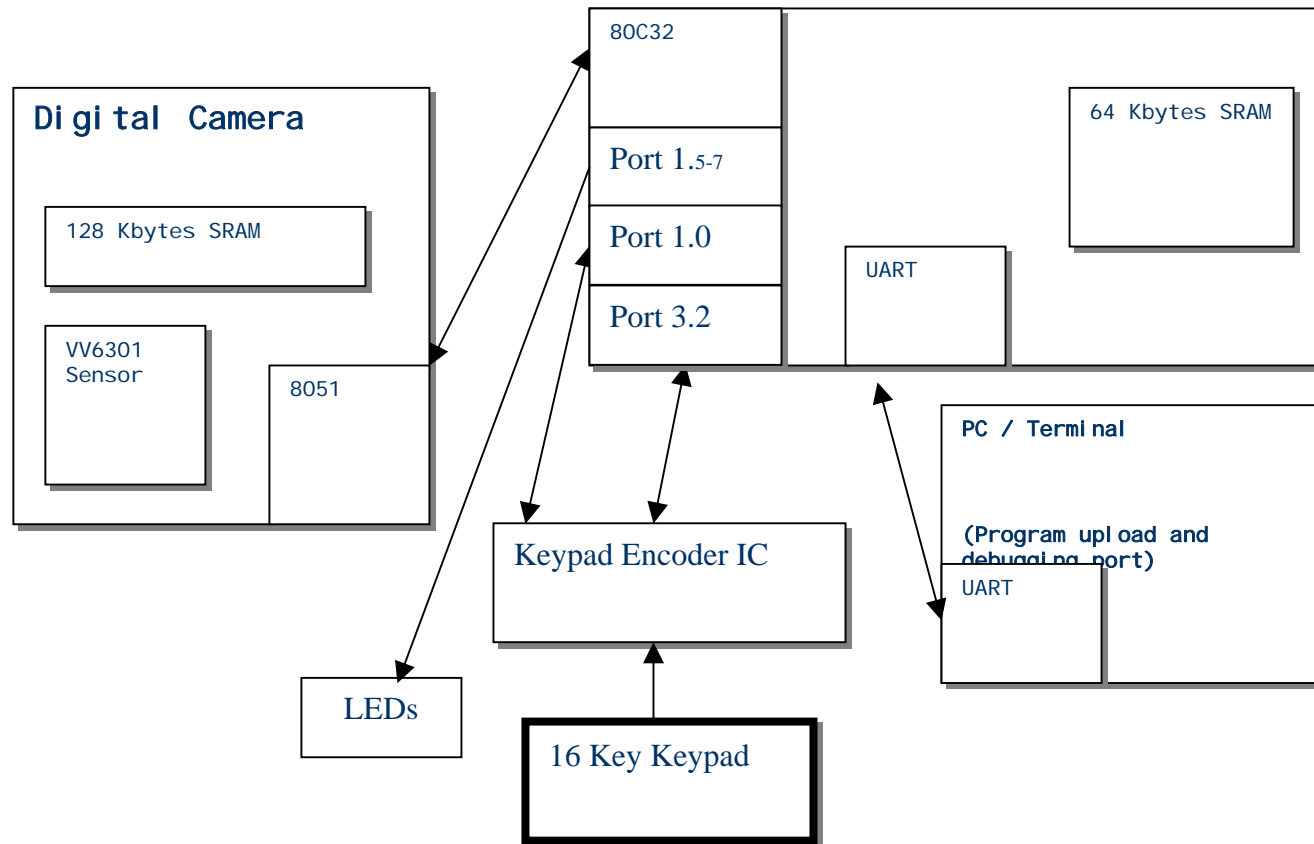
# Introduction

- This project demonstrates how to interface a low-resolution color camera with the 8032SDK,
- how to connect a 16-key keypad,
- how to capture, download and compare snapshots in less than a second, and
- how to recover the *Motion Detection System* after a sudden camera failure.

# Topics to discuss

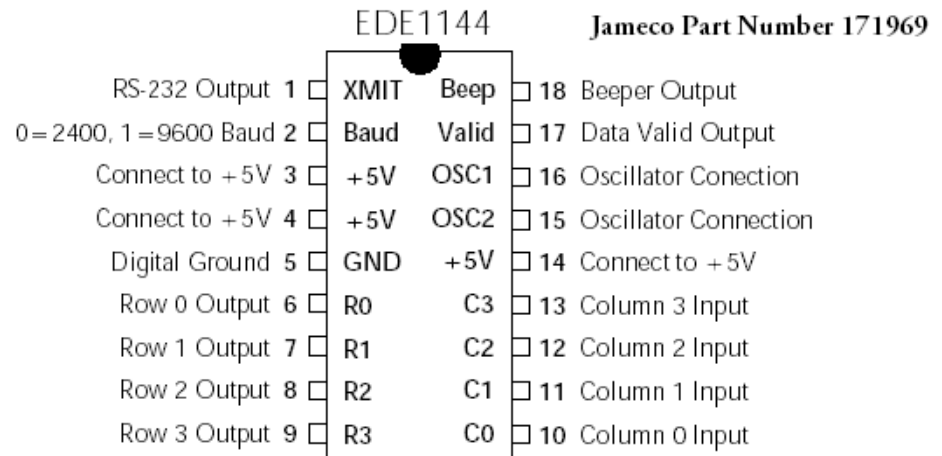
- EDE1144 keypad encoder IC
- *“High Speed”* Serial Communication
- Bayer Colorization Pattern
- Motion Detection
- Failure Recovery Strategies.

# Hardware

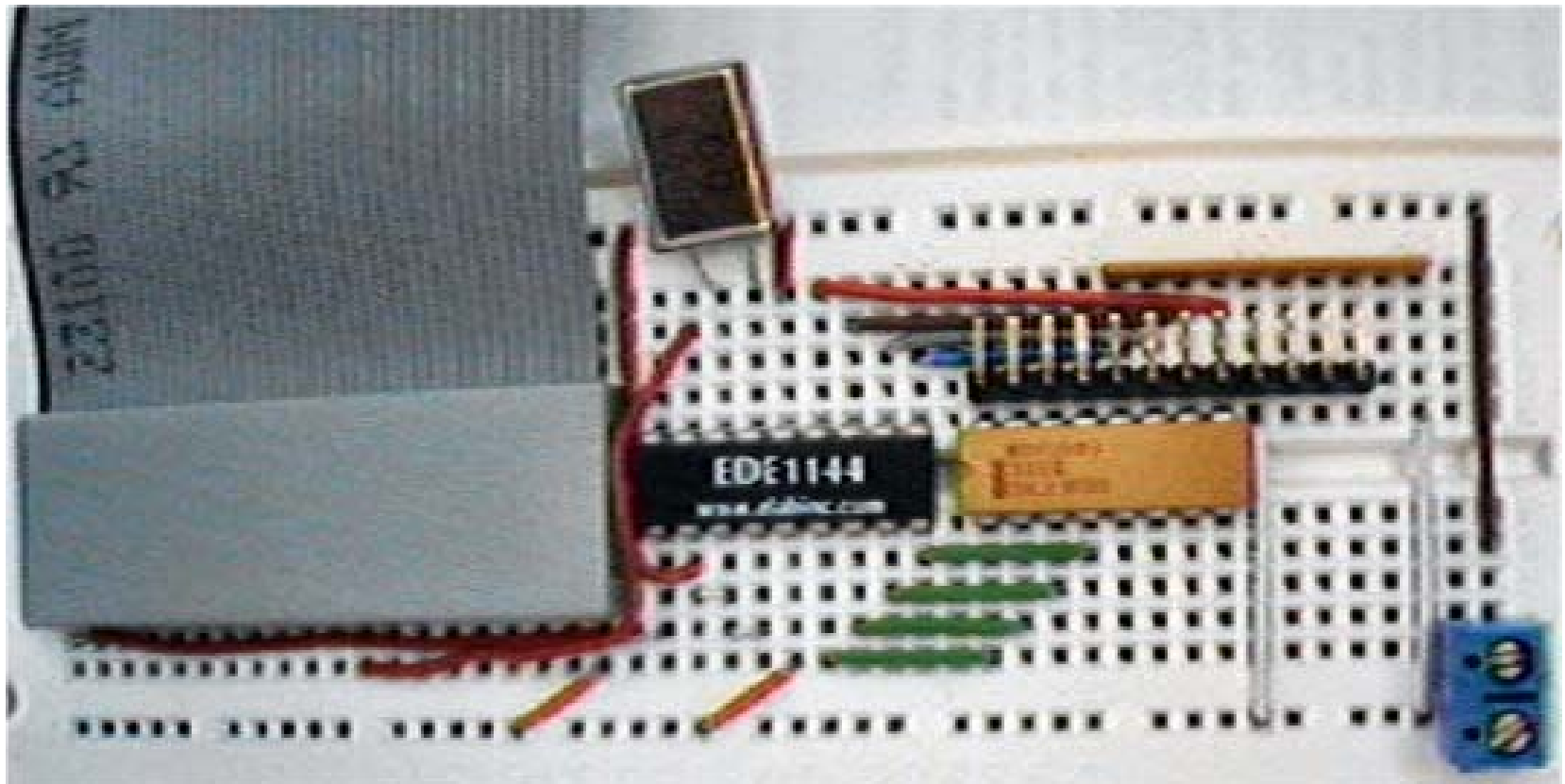


# EDE1144 keypad encoder IC

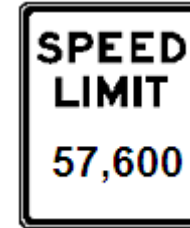
- EDE1144 keypad encoder IC interfaces a 16-key (2 rows 8 columns) keypad to the SDK's 8032 micro controller.
- Reduces the number of required I/O Pins.
- Handles contact de-bouncing.
- Provides interrupt signal



# Encoder IC, Oscillator, Resistors ..



# Camera Communication



- Timer Calculation:
  - $11.0592\text{MHz} / (12 * 57,600) = 16$
  - 16 cycles to bit-bang a port, rotate the bit into data-register or copy register value to external memory, check timeout condition
- Timeout:
  - Use Timer-1 in 16-bit auto reload mode (?!)
  - $(65536 * 12 * 14) / 11.0592\text{MHz} = 0.99555 \text{ sec}$

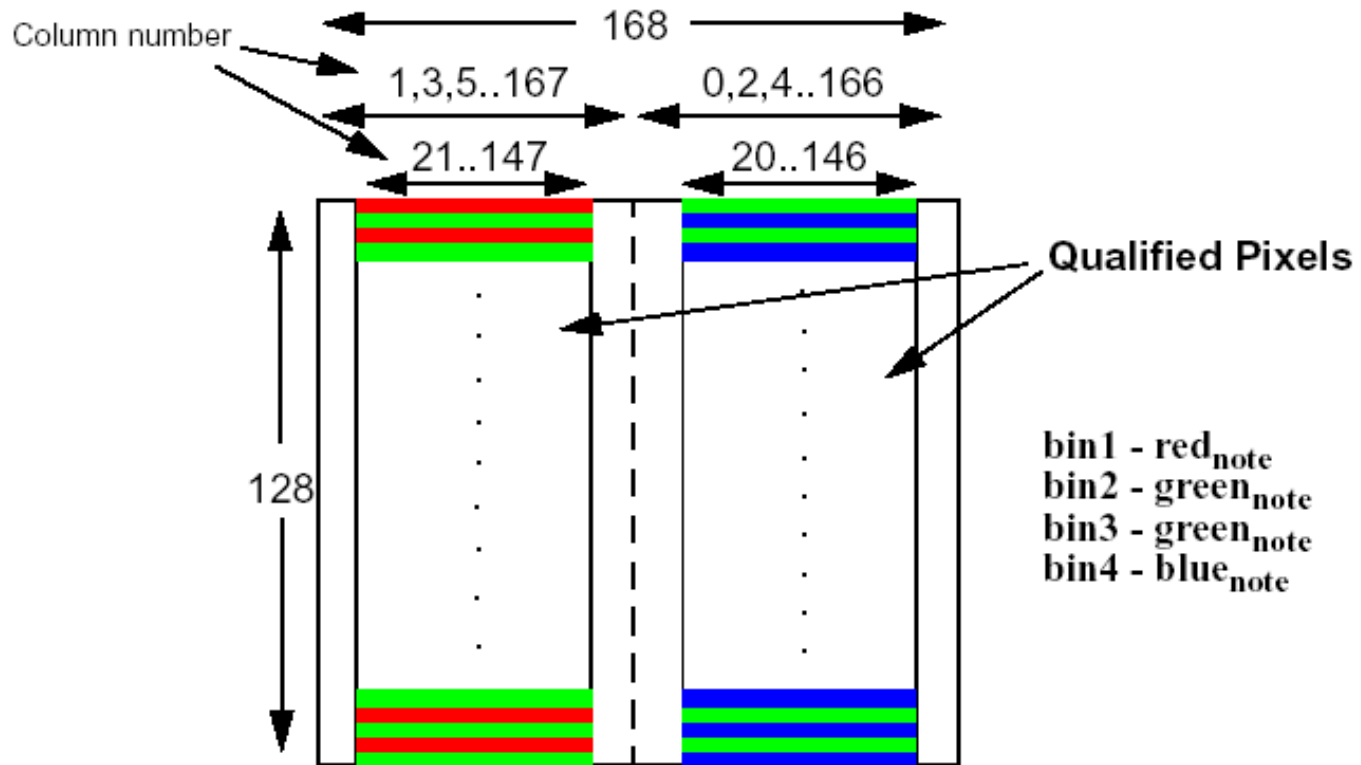
# Camera Driver

- A command from the controller has the following format:
  - *STX Command Data1 Data2...Data(n) ETX*
- The response has the following format:
  - *STX Response Data1 Data2...Data(n) ETX*
- *Example: Grab Image:*
  - *Command: 0x02 G 0x03*
  - *Response: 0x02 g 0x03*

# Bayer Pattern

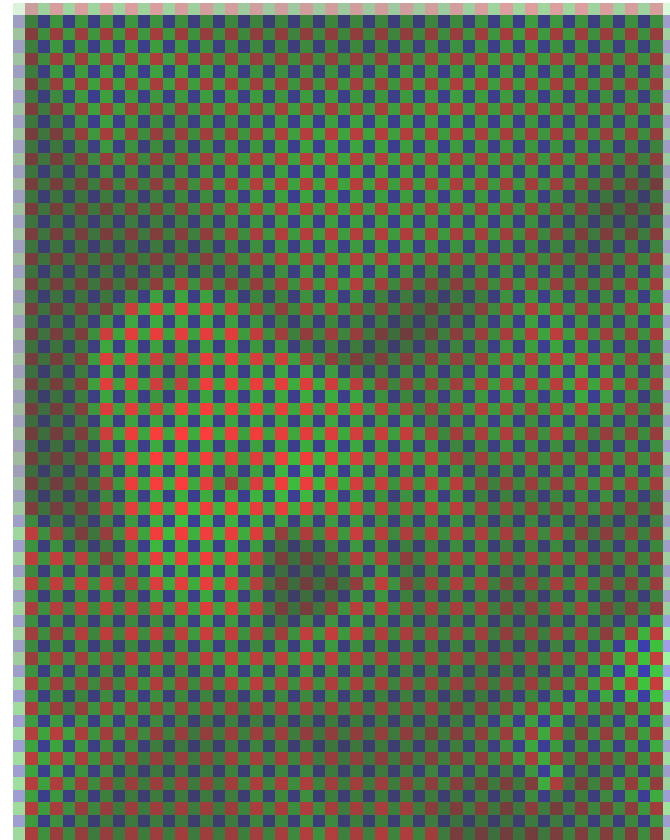
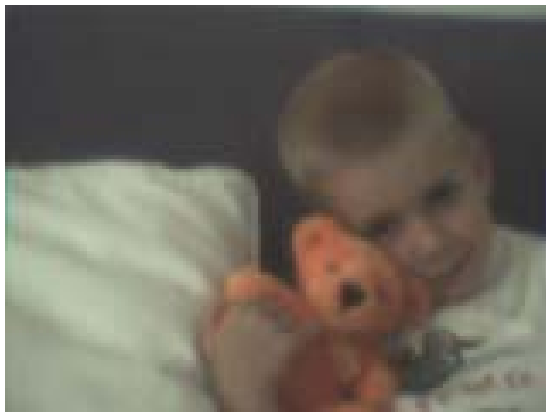


# Horizontally shuffled ...



Shuffled Pixel Readout

# Bayer-Decoding Example

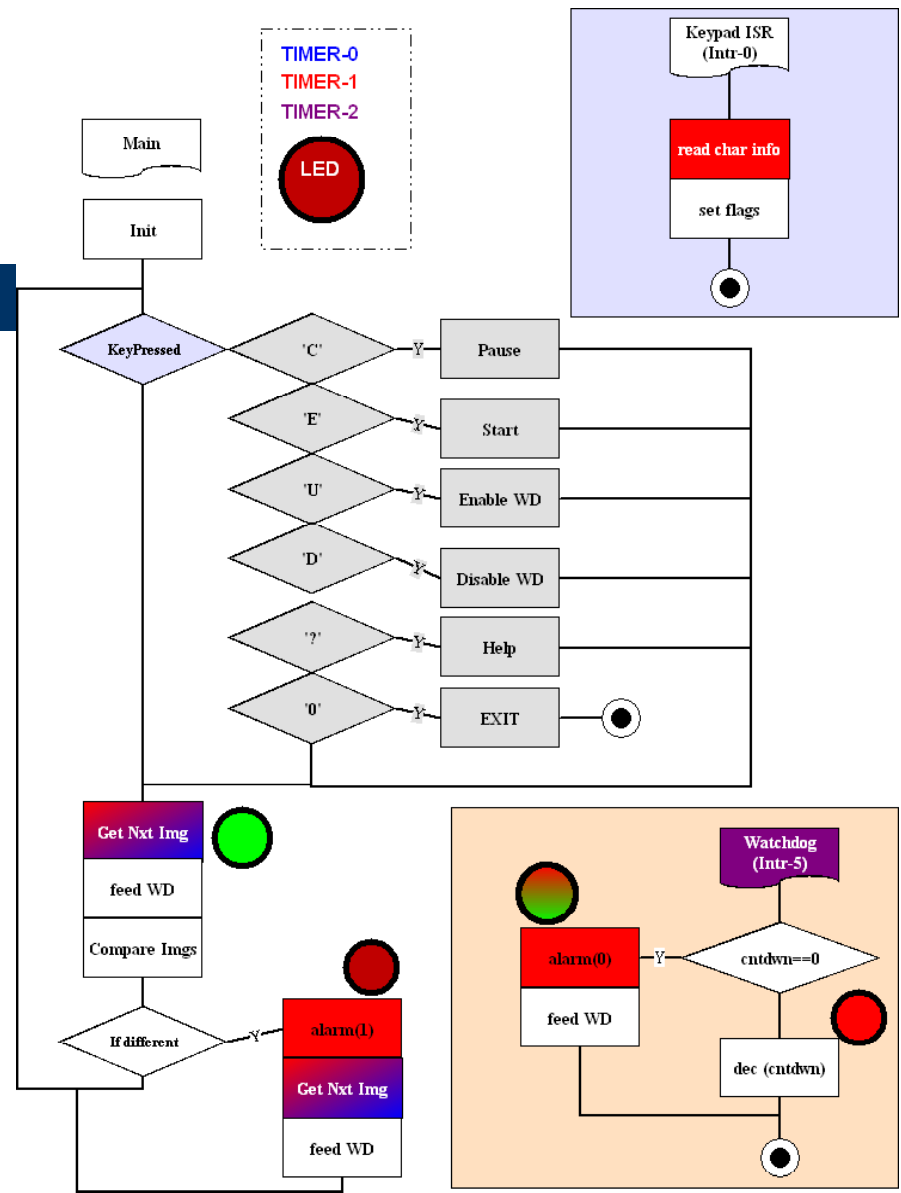




# Motion Detection, the main task ...

- Let the camera take a picture
- Download a “thumbnail” (40x31) image
- Compare this image with one, previously taken
- Execute an alert function if the difference calls for it.

# Flow Diagram



## Recovery Strategy

```
static void keypad_isr (void) interrupt 0 using 1
```

- The *initKeypad* function sets the Interrupt-0 latch high and sets it into transition (edge sensitive) mode. The Keypad-Encoder IC, for 50mS pulls the pin low; therefore the edge sensitive mode has to be used.
- After an interrupt is captured by this ISR, Timer-1 is used to *bit-bang* Port1-Pin-0 at 9600 baud, until the data byte, encoding the key that was pressed is received.

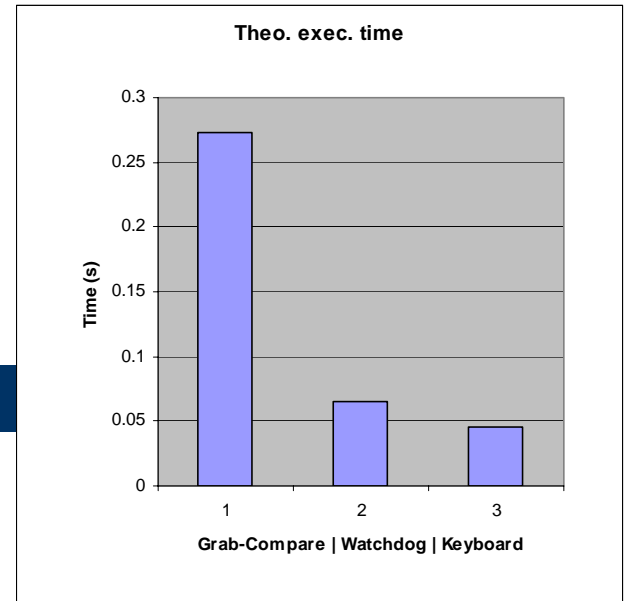
```
static void watchdog_isr(void) interrupt 5;
```

- This ISR calls the public *disableWatchdog()* function and the externally defined *alarm()* function. Therefore, this ISR uses the default R-Register set.
- The watchdog ISR only calls the alarm function if the module-static countdown counter reaches 0. However, for debugging and demonstration purposes, an LED is toggled every time the ISR executes.

# System Resources Table

Resource	Comment	Function / Program	Module
Interrupt0	External Interrupt0	keypad_isr()	KeypadDriver.C
Interrupt1	Timer-0 Interrupt	----	----
Interrupt2	External Interrupt1	---- (8031 Monitor)	---
Interrupt3	Timer-1 Interrupt	----	----
Interrupt4	Serial Interrupt	----	----
Interrupt5	Timer-2 Interrupt	watchdog_isr()	Watchdog.C
Timer 0	Timer 0	sendserial (), readserial()	serialin57.asm, serialout57.asm
Timer 1	Timer 1	<ul style="list-style-type: none"> <li>• keypad_isr()</li> <li>• readserial()</li> <li>• alarm()</li> </ul>	<ul style="list-style-type: none"> <li>• KeypadDriver.C</li> <li>• serialin57.asm</li> <li>• main.c</li> </ul>
Timer 2	Timer 2	watchdog_isr()	Watchdog.C

# Recovery and Timing



Grab-Compare Freq.	sec	cycles	Watchdog	sec	cycles	Keypad	sec	cycles
<b>Get Nxt Img</b>						<b>Event</b>		
sendSerial(4)	0.0008	800						
readSerial(4)	0.0008	800						
sendSerial(4)	0.0008	800						
readSerial(1240)	0.2	198400						
	0.2024	200800	Triggered	0.065	<b>65536</b>			
<b>feed WD</b>						<b>Pause</b>	0.045	45000
<b>Cmp Img</b>	0.07	70000				<b>Bit bang</b>	0.0009	900
Theoretical limit	0.2724	270800		0.065	<b>65536</b>		0.0459	45900
Actual behavior	0.6							

# DEMO

Keypad Label	EDE1144 Output	Lookup Table	Comments
1	0x30 (48d)	'1'	
2	0x31 (49d)	'2'	
3	0x32 (50d)	'3'	
↑	0x33 (51d)	'U'	Enable Watchdog
4	0x34 (52d)	'4'	
5	0x35 (53d)	'5'	
6	0x36 (54d)	'6'	
↓	0x37 (55d)	'D'	Disable Watchdog
7	0x38 (56d)	'7'	
8	0x39 (57d)	'8'	
9	0x41 (65d)	'9'	
2 <sup>nd</sup>	0x42 (66d)	'S'	
Clear	0x43 (67d)	'C'	Pause
0	0x44 (68d)	'0'	Exit
Help	0x45 (69d)	'?'	Help
Enter	0x46 (70d)	'E'	Start / Continue

