

# 24LC128 (#602-00013)

## General Description

The 24LC128 is a 128 Kb (16K X 8) I<sup>2</sup>C CMOS Serial EEPROM that can provide additional non-volatile data storage for the BASIC Stamp<sup>®</sup> 2 Modules. The 8-pin DIP package is easy to interface to the BS2, BS2e, BS2sx, BS2p24/40, BS2pe and BS2px24.

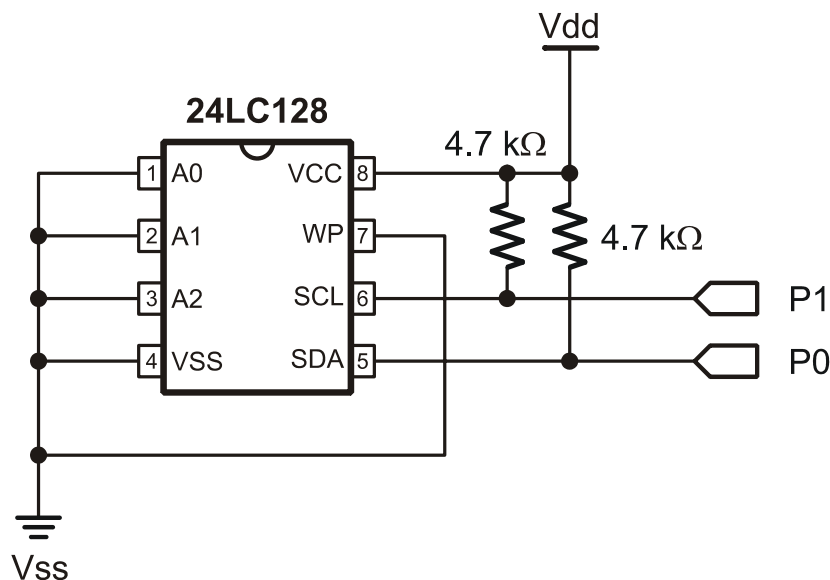
## Features

- Low-power CMOS technology
- Maximum write current 3 mA at 5.5 V
- Maximum read current 400  $\mu$ A at 5.5 V
- Standby current 100 nA typical at 5.5 V
- 5 mS max write cycle time
- 1,000,000 erase/write cycles
- 2-wire serial interface, I<sup>2</sup>C<sup>™</sup> compatible
- 8-pin .300" DIP Package
- Compatible with all BASIC Stamp 2 and SX Modules

## Application Ideas

- Data storage
- Data tables

## Quick Start Circuit



## Connecting and Testing

Connect the 24LC128 as shown in the Quick Start Circuit, then use the appropriate demo program for your BASIC Stamp module to verify correct operation. If you are using an SX Microcontroller you will need to write or edit some existing I<sup>2</sup>C code to access the EEPROM.

## Resources and Downloads

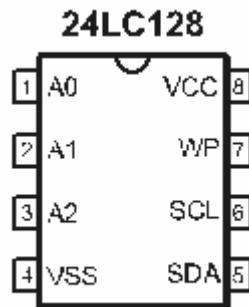
Check out the 24LC128 product page for example programs, articles and more:

[http://www.parallax.com/detail.asp?product\\_id=602-00013](http://www.parallax.com/detail.asp?product_id=602-00013)

## Device Information

### Pin Definitions and Ratings

Pin	Name	Function
1	A0	Address Pin 0
2	A1	Address Pin 1
3	A2	Address Pin 2
4	Vss	Ground -> 0 V
5	SDA	Serial Data Input/Output
6	SCL	Synchronous Clock Input
7	WP	Write-Protect Pin
8	Vcc	+5 V Power Input



Symbol	Quantity	Minimum	Typical	Maximum	Units
Vcc	Supply Voltage	2.5	5.0	6.5	V
GND	Ground		0		V
I <sub>cc</sub> (sb)	Standby Supply Current		100		nA
I <sub>r</sub>	Read Current		400		uA
I <sub>w</sub>	Write Current		3		mA

\* Ratings from manufacturer datasheet

## Communication Protocol

The 24LC128 communicates over the I<sup>2</sup>C bus with various types of transactions, such as Byte Write, Page Write, Random Read and Sequential Read.

All communication with the 24LC128 requires a control byte to be sent. This control byte includes a Start Bit, the Control Code (4 bits), the Chip Select (3 bits) and the R/W bit.

The address lines (A0, A1, A2) are used to select multiple 24LC128 devices on the same bus. Up to 8 24LC128 devices can be connected to the same bus.

Please see the manufacturer datasheet from Microchip for more information on reading/writing data to the 24LC128.

## Source Code

### BASIC Stamp<sup>®</sup> 2, 2e, 2sx Program

This program will allow you to use the 24LC128 with the BASIC Stamp 2, 2e and 2sx.

```
' =====
' File..... 24LC128.bs2
' Purpose.... Interface To A 24LC128 16KB EEPROM
' Author..... Parallax, Inc.
' E-mail..... support@parallax.com
' Updated.... 08-09-2005
' {$STAMP BS2}
' {$PBASIC 2.5}
' -----[ Program Description ]-----

' This code will interface the BS2, BS2e and BS2sx to a 24LC128 16KB EEPROM

' -----[ I/O Definitions ]-----

SDA          PIN      0          ' I2C Serial Data Line
SCL          PIN      1          ' I2C Serial Clock Line

' -----[ Constants ]-----

Address      CON      $A0        ' Address Of 24LC128
Ack          CON      0          ' Acknowledge Bit

' -----[ Variables ]-----

temp         VAR      Byte       ' Variable Used For Work
i2cWork      VAR      Word       ' Work Byte For I2C I/O
i2cAck       VAR      Bit        ' Ack Bit From Device
eeprom_add   VAR      Word       ' EEPROM Address

' -----[ Main Routine ]-----

DO
  DEBUG "Enter value to store from 0 to 255: "
```

```

DEBUGIN DEC3 temp
DEBUG CR, "Enter address to store ", DEC temp, " at, 0 to 15999: "
DEBUGIN DEC5 eeprom_add
' Writing Section
GOSUB I2C_Stop
GOSUB I2C_Start
GOSUB Control_Byte_Write
GOSUB Addr
GOSUB Write_Data
GOSUB I2C_Stop
PAUSE 100
' Reading Section
GOSUB I2C_Start
PAUSE 10
GOSUB Control_Byte_Write
GOSUB Addr
GOSUB I2C_Start
GOSUB Control_Byte_Read
SHIFTIN SDA, SCL, LSBFIRST, [i2cwork\8] ' Send Byte To Device
GOSUB I2C_Stop
DEBUG CR, "Value stored at location ", DEC eeprom_add,
" is ", DEC i2cwork.LOWBYTE, CR, CR
LOOP

' -----[ Subroutines ]-----

I2C_Stop:                                ' I2C Stop Bit Sequence
LOW SDA
INPUT SCL
INPUT SDA                                ' SDA --> High While SCL High
RETURN

I2C_Start:                                ' I2C Start Bit Sequence
INPUT SDA
INPUT SCL
LOW SDA                                  ' SDA --> Low While SCL High
RETURN

Addr:                                     ' I2C Address Byte Sequence
i2cWork = eeprom_add
SHIFTOUT SDA, SCL, MSBFIRST, [i2cWork.HIGHBYTE\8] ' Send Byte To Device
SHIFTIN SDA, SCL, MSBPRES, [i2cAck\1] ' Get Acknowledge Bit
SHIFTOUT SDA, SCL, MSBFIRST, [i2cWork.LOWBYTE\8] ' Send Byte To Device
SHIFTIN SDA, SCL, MSBPRES, [i2cAck\1] ' Get Acknowledge Bit
RETURN

Control_Byte_Write:                       ' I2C Control Write Byte Sequence
i2cWork = Address
i2cWork.BIT0 = 0                          ' Sets Unit To Write
SHIFTOUT SDA, SCL, MSBFIRST, [i2cWork\8] ' Send Byte To Device
SHIFTIN SDA, SCL, MSBPRES, [i2cAck\1] ' Get Acknowledge Bit
RETURN

Control_Byte_Read:                        ' I2C Control Read Byte Sequence
i2cWork = Address

```

```

i2cWork.BIT0 = 1          ' Sets Device To Read
SHIFTOUT SDA, SCL, MSBFIRST, [i2cWork\8] ' Send Byte To Device
SHIFTIN SDA, SCL, MSBPRES, [i2cAck\1] ' Get Acknowledge Bit
RETURN

Write_Data:              ' I2C Write Byte Sequence
i2cWork = temp
SHIFTOUT SDA, SCL, MSBFIRST, [i2cWork\8] 'Send Byte To Device
SHIFTIN SDA, SCL, MSBPRES, [i2cAck\1] ' Get Acknowledge Bit
RETURN

```

## BASIC Stamp® 2p, 2pe, 2px Program

This program will allow you to use the 24LC128 with the BASIC Stamp 2p, 2pe and 2px.

```

' =====
' File..... 24LC128.bsp
' Purpose.... Interface To A 24LC128 16KB EEPROM
' Author..... Parallax, Inc.
' E-mail..... support@parallax.com
' Updated.... 08-09-2005
' {$STAMP BS2p}
' {$PBASIC 2.5}

' -----[ Program Description ]-----
' This code will interface the BS2p, BS2pe and BS2px to a 24LC128 EEPROM

' -----[ I/O Definitions ]-----
SDA          PIN      0          ' I2C Serial Data Line
SCL          PIN      1          ' I2C Serial Clock Line

' -----[ Variables ]-----
value        VAR      Byte      ' Variable Used To Store Value
eeprom_add   VAR      Word      ' EEPROM Address

' -----[ Main Routine ]-----
DO
  DEBUG "Enter value to store from 0 to 255: "
  DEBUGIN DEC3 Value
  DEBUG CR, "Enter address to store ", DEC value, " at, 0 to 15999: "
  DEBUGIN DEC5 eeprom_add
  ' Writing Section
  I2COUT SDA, $A0,eeprom_add.HIGHBYTE\eeprom_add.LOWBYTE, [value]

  PAUSE 20

  ' Reading Section
  I2CIN SDA, $A1,eeprom_add.HIGHBYTE\eeprom_add.LOWBYTE, [value]
  DEBUG CR, "Value stored at location ",DEC eeprom_add,
    " is ", DEC value, CR, CR, CR
LOOP

```