stamp.peripheral.memory.eeprom

Class MC24LC32LibEx

```
java.lang.Object
  |
  +--stamp.peripheral.memory.eeprom.MC24LC32LibEx
```

public class **MC24LC32LibEx**
extends Object

This class is for demonstration purposes only. It should be used in conjunction with Javelin Stamp Application Note 3: I2C Primer - EEPROM example.

This class can be instantiated for each 24LC32 on a given I2C bus, and it contains methods that enable bytewise and multi-byte read and write operations.

Constructor Summary

MC24LC32LibEx(I2C i2cbus, int chipAddress) Create MC24LC32 object by passing an I2C bus and the 24LC32's chip address to this constructor.

Method Summary

int	**readByte**(int eeAddress) Read a byte value from an address in the 24LC32.
void	**readStringIntoBuffer**(int eeAddress, int count, StringBuffer sb) Read a string of characters of a specific length starting at a paricular address in the 24LC32.
void	**setAddress**(int eeAddress) Set the 24LC32's EEPROM address pointer.
void	**writeByte**(int eeAddress, int dataByte) Write a byte value to a particular address in the 24LC32.
void	**writeStringToEeprom**(int eeAddress, StringBuffer sb) Write a string of characters starting at a particular address in the 24LC32.

Methods inherited from class java.lang.Object
equals

Constructor Detail

MC24LC32LibEx

```
public MC24LC32LibEx(I2C i2cbus,
                     int chipAddress)
```

Create MC24LC32 object by passing an I2C bus and the 24LC32's chip address to this constructor. For example:

```
// Create an I2C bus object named i2cbus. final public static int SDAPin =
CPU.pin6; final public static int SCLPin = CPU.pin7; public static I2C
i2cbus = new I2C(SDAPin, SCLPin); // Create a Microchip24LC32 object named
eeprom0 using the i2cbus object. public static Microchip24LC32 eeprom0 = new
Microchip24LC32(i2cbus, 0);
```

Parameters:

`i2cbus` - the I2C bus object that has the new 24LC32 object/chip connected to it.
`chipAddress` - the binary address value of the new 24LC32 chip. This should be the binary value of A2, A1, A0.

Method Detail

setAddress

```
public void setAddress(int eeAddress)
```

Set the 24LC32's EEPROM address pointer.

Parameters:

`eeAddress` -

---

writeByte

```
public void writeByte(int eeAddress,
                      int dataByte)
```

Write a byte value to a particular address in the 24LC32.

Parameters:

`eeAddress` - the address where the byte value should be stored.
`dataByte` - the byte value to be stored at `eeAddress`.

---

readByte

```
public int readByte(int eeAddress)
```

Read a byte value from an address in the 24LC32.

Parameters:
`eeAddress` - the address that contains the byte to be read.
Returns:
value the byte stored at `eeAddress`.

---

writeStringToEeprom

```
public void writeStringToEeprom(int eeAddress,
                                StringBuffer sb)
```

Write a string of characters starting at a particular address in the 24LC32.

Parameters:
`eeAddress` - the address where the byte value should be stored.
`sb` - the `StringBuffer` object that contains the string of characters.

---

readStringIntoBuffer

```
public void readStringIntoBuffer(int eeAddress,
                                 int count,
                                 StringBuffer sb)
```

Read a string of characters of a specific length starting at a paricular address in the 24LC32.

Parameters:
`eeAddress` - the starting address at the beginning of the string.
`count` - the number of characters to read
`sb` - the `StringBuffer` object that stores the string of characters.
