

```
//for use with Techspace Learning Shift Register module.
//Displays 0 - 99 on two 7-seg displays using bytes and bitreading.
```

```
int dataIn = 4;
int klock = 7;
int latch = 6;
int bitNum = 0;
int decNum = 0;
int units;
int tens;
int enable = 5; //PWM pin
int pot = A0;
int potValue;
```

```
//bytes converted to Hexa
```

```
byte numerals[] {
```

```
  0x7E, //ZERO
  0x30, //ONE
  0x6D, //TWO
  0x79, //THREE
  0x33, //FOUR
  0x5B, //FIVE
  0x5F, //SIX
  0x70, //SEVEN
  0x7F, //EIGHT
  0x7B, //NINE
};
```

```
void setup() {
```

```
  pinMode(dataIn,OUTPUT);
  pinMode(klock,OUTPUT);
  pinMode(latch,OUTPUT);
  pinMode(enable,OUTPUT);
  pinMode(pot,INPUT);
}
```

```
void Klocking() {
```

```
  digitalWrite(klock,1);
  digitalWrite(klock,0);
}
```

```
void Latch() {
```

```
  digitalWrite(latch,1);
  digitalWrite(latch,0);
}
```



```
void loop() {
```

```
  potValue = analogRead(pot);
  potValue = potValue/4;
  analogWrite(enable,potValue); //Inverse logic.
```

```
  tens = decNum/10;
  units = decNum%10;
```

```
//Using modulo to find remainder.
```

```
  for(bitNum = 0;bitNum < 8;bitNum++){
```

```
    digitalWrite(dataIn,bitRead(numerals[tens],bitNum));
    Klocking();
  }
```

```
  for(bitNum = 0;bitNum < 8;bitNum++){
```

```
    digitalWrite(dataIn,bitRead(numerals[units],bitNum));
    Klocking();
  }
```

```
  Latch();
  delay(500);
```

```
  decNum++;
  if(decNum > 99) {
    decNum = 0;
  }
}
```

