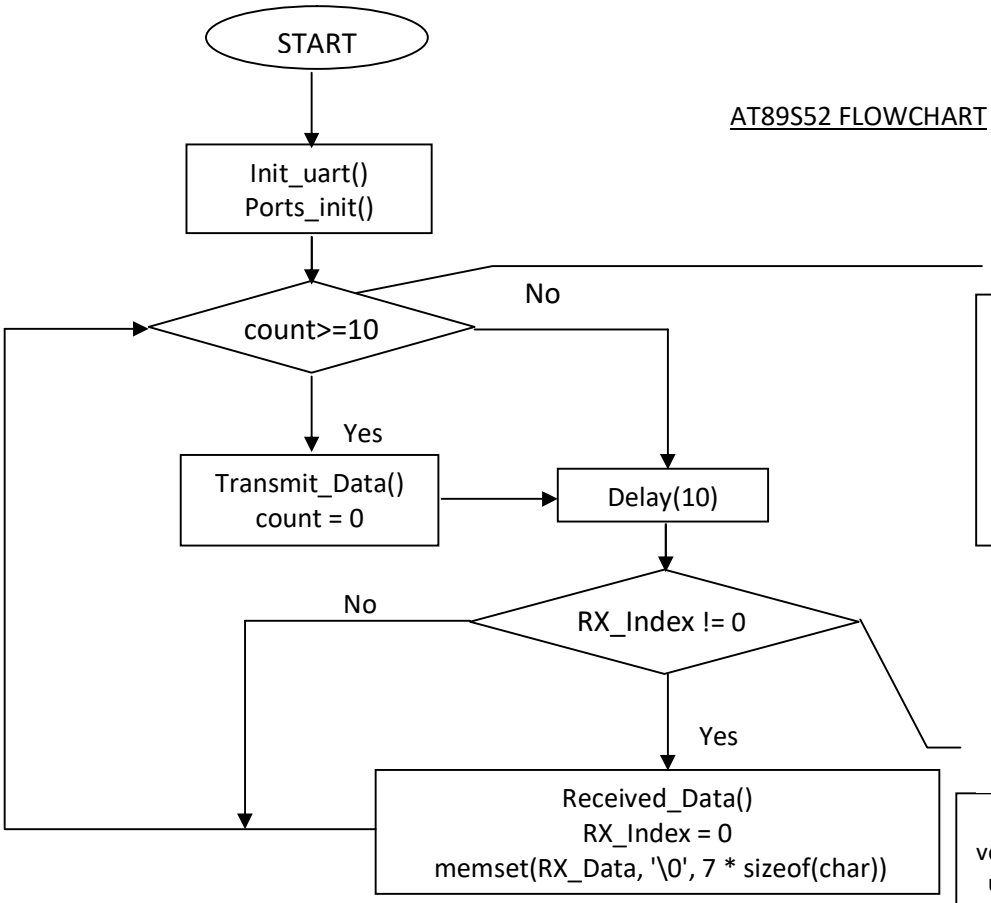

AT89S52 FLOWCHART

"count" value based on timer

Timer Interrupt

```
void timer0_isr() interrupt 1 {
  TH0 = 0x4B;
  TL0 = 0xFD;
  count++;
}
```

"RX_Index" value based on uart interrupt

Uart Interrupt

```
void serial_isr() interrupt 4 {
  unsigned char receivedChar;
  if (RI == 1) {
    RI = 0; // Clear the Receive interrupt flag
    receivedChar = SBUF;
    //if (RX_Index >= 6) RX_Index=0;
    //RX_Data[RX_Index++] = receivedChar;
    if (RX_Index >= RST_RX_Index) RX_Index = 0;
    else {
      RX_Data[RX_Index++] = receivedChar;
      if (RX_Data[0] != 0xA5) RX_Index = 0;
    }
  }
}
```

Received Data()

```
 if (RX_Data[0] == IDENTITY_BYTE) {
   if (RX_Data[1] == START_BYTE) {
     unsigned char temp = RX_CHECKSUM(RX_Data[2], RX_Data[3],
RX_Data[4]);
     if ((RX_Data[5] == temp) && (RX_Data[6] == END_BYTE)) {
       Rx_012Bit(&RX_Data[2]);
       Rx_345Bit(&RX_Data[2]);
                                    if(RX_Data[4]>=3)
                                              (PORT3_PIN4=0);
                                    else
                                              (PORT3_PIN4=1);
                }
            }
       }
}
```

Transmit Data()

```
 TX_Data[0] = IDENTITY_BYTE;
 TX_Data[1] = START_BYTE;
 TX_Data[2] = ~P0;
 TX_Data[3] = ~P2;
 TX_Data[4] = TX_CHECKSUM(TX_Data[2],
TX_Data[3]);
 TX_Data[5] = END_BYTE;
 Tx_char(TX_Data[0]);
 Tx_char(TX_Data[1]);
 Tx_char(TX_Data[2]);
 Tx_char(TX_Data[3]);
 Tx_char(TX_Data[4]);
 Tx_char(TX_Data[5]);
```