```
#include <EEPROM.h>

int time = 0;
int segmentPins[] = { 6, 7, 8, 9, 10, 11, 12, 13 };  // 数码管的段pin
int displayPins[] = { 0, 1, 4, 5 };                  // 数码管的位pin
int timerMinute;
int timerSecond;
int buzzerPin = 15;
int aPin = 2;
int bPin = 3;
int buttonPin = 14;
int state = 0;   //0 代表调整分钟；1代表调整秒钟；2代表计时

byte digits[11][8] = {
  { 1, 1, 1, 1, 1, 1, 0, 0 },  //0
  { 0, 0, 0, 0, 1, 1, 0, 0 },  //1
  { 1, 1, 0, 1, 1, 0, 1, 0 },  //2
  { 1, 0, 0, 1, 1, 1, 1, 0 },  //3
  { 0, 0, 1, 0, 1, 1, 1, 0 },  //4
  { 1, 0, 1, 1, 0, 1, 1, 0 },  //5
  { 1, 1, 1, 1, 0, 1, 1, 0 },  //6
  { 0, 0, 0, 1, 1, 1, 0, 0 },  //7
  { 1, 1, 1, 1, 1, 1, 1, 0 },  //8
  { 1, 0, 1, 1, 1, 1, 1, 0 },  //9
  { 0, 0, 0, 0, 0, 0, 0, 0 }
};

void setup() {
  for (int i = 0; i < 8; i++)
    pinMode(segmentPins[i], OUTPUT);
  for (int i = 0; i < 4; i++)
    pinMode(displayPins[i], OUTPUT);

  pinMode(buzzerPin, OUTPUT);
  pinMode(buttonPin, INPUT);
  pinMode(aPin, INPUT);
  pinMode(bPin, INPUT);

  time = EEPROM.read(0);
  timerMinute = time / 60;
  timerSecond = time % 60;
}

void loop() {
  if (!digitalRead(buttonPin)) {
    state++;
    state = (state > 2 ? 0:2);
    digitalWrite(buzzerPin, LOW);
    while (!digitalRead(buttonPin)) {};
    EEPROM.write(0, time);
  }
  updateDisplay();
}

void updateDisplay()  //mmss
{
  int minsecs = timerMinute * 100 + timerSecond;
  int v = minsecs;
  for (int i = 0; i < 4; i++) {
    int digit = v % 10;
    setDigit(i);
    setSegments(digit);
    v = v / 10;
    process();
    setSegments(10);
  }
  setDigit(5);
}

void process() {
  for (int i = 0; i < 100; i++)  //tweak this number between flicker and blur
  {
    int change = getEncoderTurn();
    if (state == 0)
      changeSetMinte(change);
    else if (state == 1)
      changeSetSecond(change);
    else if (state == 2)
      updateCountingTime();
  }
  if (timerMinute == 0 && timerSecond == 0)
    digitalWrite(buzzerPin, HIGH);
}

void changeSetSecond(int change) {
  time += change;
 if(time < 0 && state == 1)
    time += 6040;
  timerMinute = time / 60;
  timerSecond = time % 60;
}

void changeSetMinte(int change) {
  time += change * 60;
  if(time < 0 && state == 0)
    time += 6040;
  timerMinute = (time / 60) % 100;
  timerSecond = time % 60;
}

void updateCountingTime() {
  static unsigned long lastMillis;
  unsigned long m = millis();
  if (m > (lastMillis + 1000) && (timerSecond > 0 || timerMinute > 0)) {
    digitalWrite(buzzerPin, LOW);
    delay(10);
    digitalWrite(buzzerPin, LOW);

    if (timerSecond == 0) {
      timerSecond = 59;
      timerMinute--;
    } else
      timerSecond--;
    lastMillis = m;
  }
  if (timerMinute == 0 && timerSecond == 0){
    digitalWrite(buzzerPin, HIGH);
    timerMinute = 0;
    timerSecond = 0;
  }


}

void setDigit(int digit) {
  for (int i = 0; i < 4; i++) {
    int on = (digit == i) ? HIGH : LOW;
    digitalWrite(displayPins[i], on);
  }
}

void setSegments(int n) {
  for (int i = 0; i < 8; i++) {
    digitalWrite(segmentPins[i], digits[n][i]);
  }
}

int getEncoderTurn() {
  // return -1, 0, or +1
  static int oldA = LOW;
  static int oldB = LOW;
  int result = 0;
  int newA = digitalRead(aPin);
  int newB = digitalRead(bPin);
  if (newA != oldA || newB != oldB) {
    // something has changed
    if (oldA == LOW && newA == HIGH)
      result = -(oldB * 2 - 1);
  }
  oldA = newA;
  oldB = newB;
  return result;
}
```

嫦娥SetScondset