

## GSM2 Click Kartı Arduino Kullanımı

Doküman Tarihi:25.11.2018\_Rev1.0

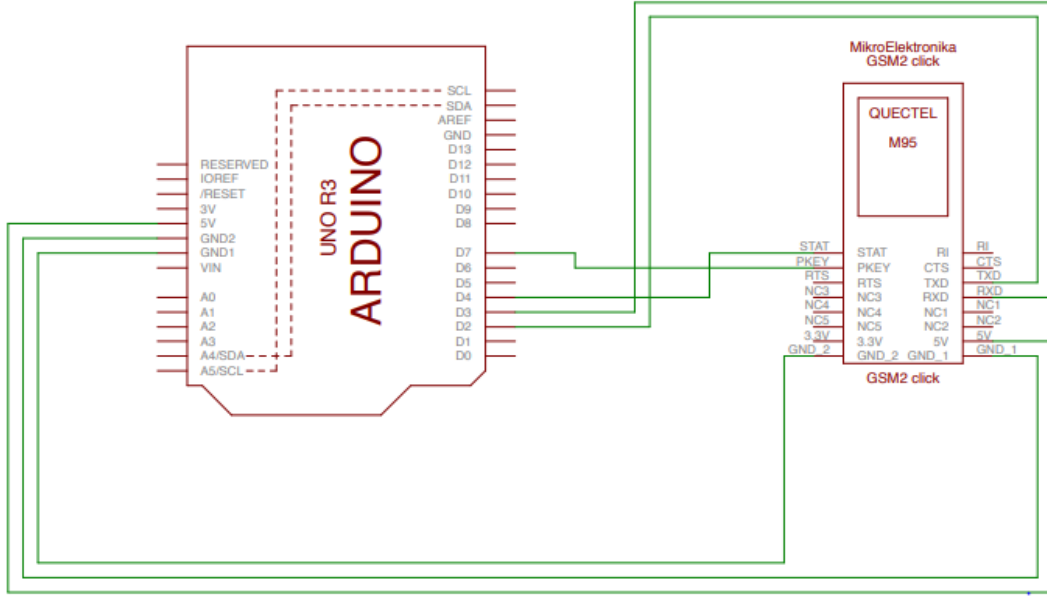


GSM2 Click, GSM / GPRS Quectel M95'i içeren kompakt bir dört bantlı GSM hücresel ağ iletişim çözümüdür. Bu modül hücresel ağ durumu gösterimi, Jamming Detection, TCP/IP, UDP, FTP, PPP, HTTP, SMTP, full GPRS multislot class 12 implementation içeren gömülü internet protokolleri gibi ağ ve haberleşmesi için bir dizi seçenek sunar. GSM Phase 2/2+, ve daha fazlası ile tamamen uyumludur. Veri haberleşme hızı, hem uplink hem de downlink bağlantısı için 85.6 kbps'ye kadar derecelendirilmiştir.

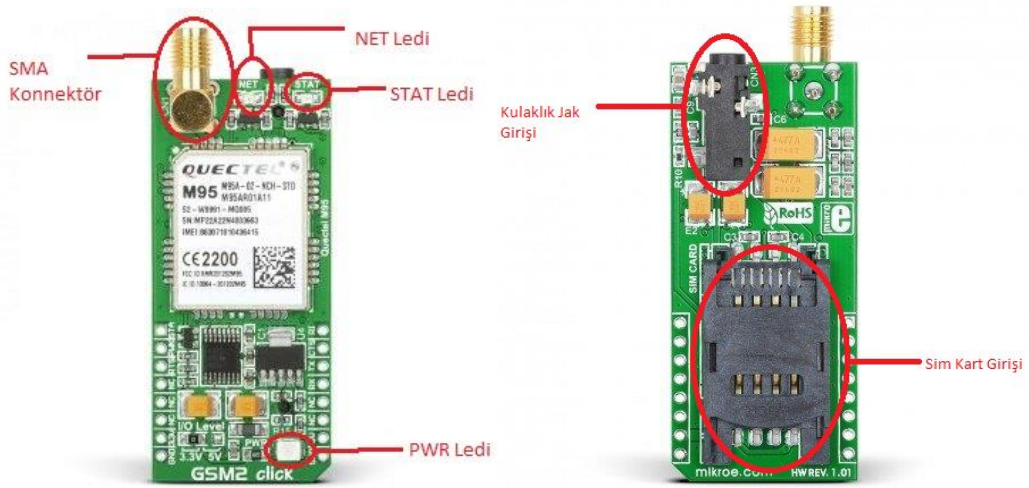
Bu doküman GSM2 Click modülünün Arduino kartında nasıl kullanılacağını ve pinlere bağlanacağını anlatmaktadır. GSM2 Click modülünü kullanmaya başlamadan önce dikkat edilmesi gereken en önemli işlem, karta takacağınız **SIM kartının pin numarasının kaldırılmış olması işinizi kolaylaştıracaktır. Bu işlemi bir telefona takıp yapabilirsiniz.**

### Pin Bağlantıları ve Modülün Kurulumu

Arduino Uno ile GSM2 Click arasındaki bağlantılar aşağıda gösterildiği gibi dikkatli ve doğru bir şekilde yapılmalıdır. Aksi takdirde modül istenilen şekilde çalışmaz. Özellikle modülde bulunan RXD ve TXD pinlerinin Arduino üzerindeki D2 (Rx) ve D3 (Tx) pinlerine, TXD 'e D2 'ye ve RXD'e D3'e gelecek şekilde, bağlanıldığından emin olun.



Şekil-1(Arduino ile GSM2 Click Bağlantısı)



Şekil-2

Modülün üstünde bulunan (Şekil-2) SMA konnektörüne GSM anteni bağlamamız gerekir.



GSM Anten

Son olarak sim kartımızı Şekil-2’de gösterilen sim kart girişine sim kartının çip kısmı aşağıya bakacak şekilde yerleştirin.

Artık modülümüz kullanılmaya hazır.

### GSM2 Click Modülüne Güç Verme:

Şekil-1’de gösterildiği gibi bağlantılar yapıldıktan sonra modül gücünü Arduino’nun 5V bacağından alacaktır ve GSM2 Click üzerindeki PWR ledi sürekli yanacaktır.

### STAT Ledinin Yanması:

Modülümüzün uyku modundan çıktığını gösterir. Bu durumda işlem yapmaya başlar.

### NET Ledinin Yanması:

NET ledi modül üzerindeki diğer led göre kırmızı renkte 2-3 saniye aralıkla yanar söner. Bu, modülümüzün GSM’e bağlandığını gösterir. Artık GSM2 Click modülümüz gelen-giden çağrı ve SMS’lere açıktır.

## QUECTEL M95 Modülü AT Komutları

Bu bölümde, Arduino arayüzünde AT komutlarının çalıştırılması anlatılacaktır. Öncelikle, Arduino arayüzünde **aşağıdaki kodu** kartımıza yüklüyoruz. Daha sonra Seri Port Ekranını açıyoruz ve seri portun sağ alt kısmında Baud Rate'mizi "57600 baud" olarak ayarlıyoruz. Bu işlemden sonra metin kutumuza "AT" yazıp gönderiyoruz.

```
#include <SoftwareSerial.h>

SoftwareSerial mySerial(2, 3); // RX, TX

void setup() {
  // Seri haberleşmenin başlaması seri porta bağlanmayı beklenmesi
  Serial.begin(57600);
  while (!Serial) {
    ; // seri porta bağlanmayı beklenmesi. Sadece yerel USB bağlantı
    noktası için geçerli
  }

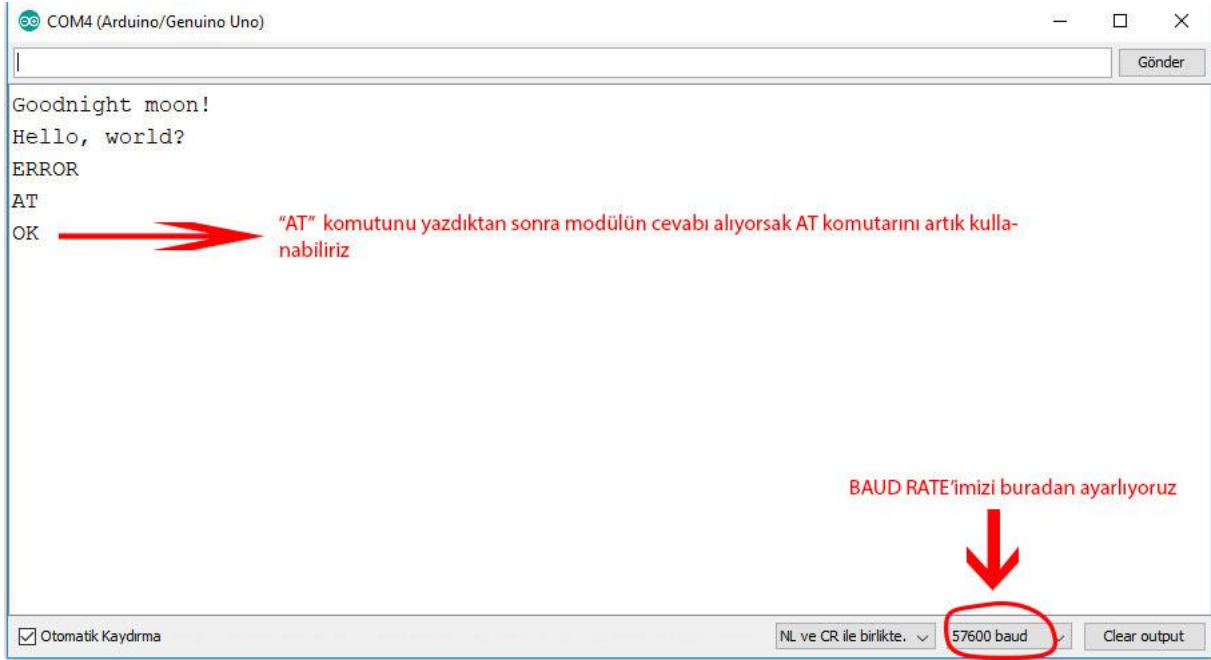
  pinMode(7, OUTPUT); // Modülün uyku modunda kalmaması ve uyku modunda
  kaldığı anda PWK pinini açık tutuyor
  digitalWrite(7, HIGH);

  Serial.println("Goodnight moon!");

  // SoftwareSerial port için veri hızını ayarlayınız
  mySerial.begin(57600);
  mySerial.println("Hello, world?");
}

void loop() {

  if (mySerial.available()) {
    Serial.write(mySerial.read()); // Modülden gelen veriyi okuyup yazması
  }
  if (Serial.available()) {
    mySerial.write(Serial.read()); // Seri porttan gelen veriyi modüle
    göndermek için
  }
}
```



Eğer “CME 100” hatası alıyorsanız,” AT” komutunu yazın “OK” cevabı aldıktan sonra “ATZ” komutunu gönderin. Sorununuz çözülmüş olacaktır.

“AT” komutları ile işlem yapmak isterseniz. [M95 FA teknik dokümanından](#) diğer komutlara ulaşabilirsiniz.

## ÖRNEK ÇALIŞMALAR

### Arduino Üzerinden GSM2 Click ile Arama Yapma

Aşağıdaki tabloda verilen Arduino kodlarını kopyalayıp Arduino Editörüne yapıştırıyoruz ve kartımıza yüklüyoruz. Daha sonra **Serial Port Ekranını** açıyoruz. Açtığımız anda kodumuz çalışmaya başlıyor. **Seri port ekranında Baud Rate ‘in kodumuzda yazdığımızla aynı olmasına dikkat ediniz. Aksi takdirde çıktı alamazsınız.**

```
#include <GSM.h>

// PIN Numarası. Eğer pin numarası varsa burada tanımlayın
#define PINNUMBER ""

// Kütüphane örneğini başlat
GSM gsmAccess; // Hata ayıklaması için "true" parametresini dahil et
GSMVoiceCall vcs;

String remoteNumber = ""; // Arayacağınız numara
char charbuffer[20];

void setup() {

    // Seri haberleşmenin başlaması seri porta bağlanmayı beklenmesi
    Serial.begin(9600);
    while (!Serial) {
        ; // seri porta bağlanmayı beklenmesi. Sadece yerel USB bağlantı
        noktası için geçerli
    }

    Serial.println("Make Voice Call");

    // Bağlantı parametresi
    boolean notConnected = true;

    // GSM shield başlat
    // Eğer PIN kodunuz varsa "begin()" parametresi içine "PINNUMBER"
    sabitini tanımlıyoruz
    while (notConnected) {
        if (gsmAccess.begin(PINNUMBER) == GSM_READY) {
            notConnected = false;
        } else {
            Serial.println("Not connected");
            delay(1000);
        }
    }

    Serial.println("GSM initialized.");
    Serial.println("Enter phone number to call.");
}
}
```

```
void loop() {

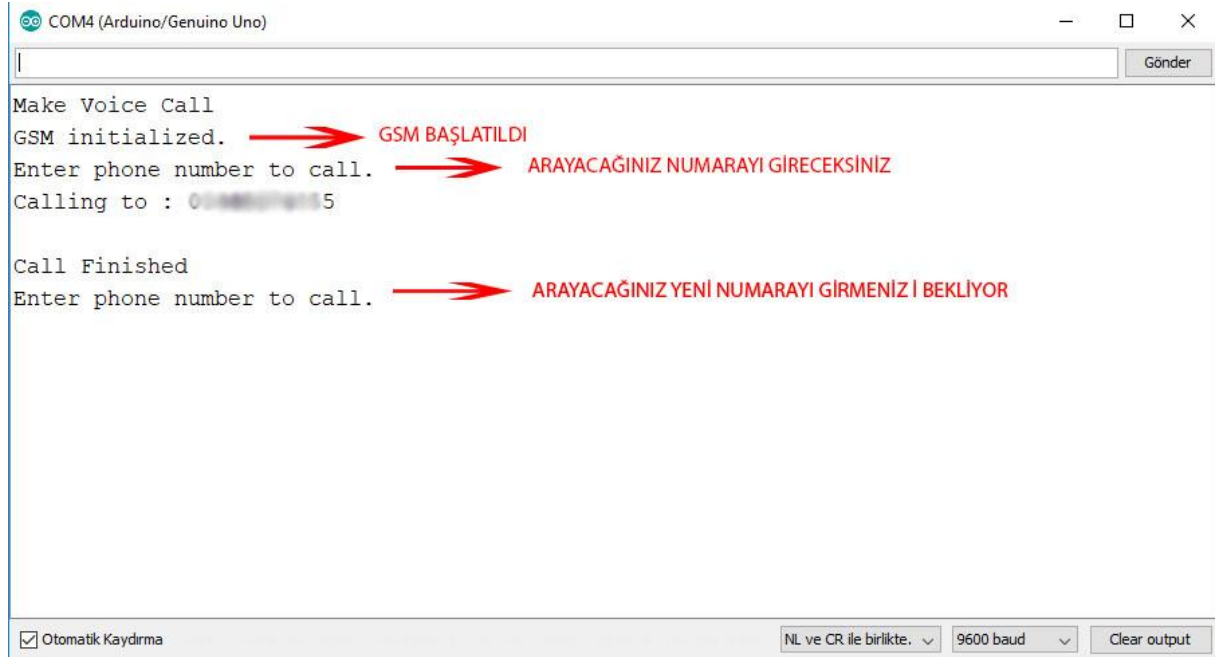
    // Gelen her karakteri String'e ekler
    while (Serial.available() > 0) {
        char inChar = Serial.read();
        // Eğer yeni bir satırsa bu aramayı yapmanız gerektiği anlamına gelir
        if (inChar == '\n') {
            // Telefon numarasının çok uzun olmaması kontrol eder
            if (remoteNumber.length() < 20) {
                // Kullanıcıyı aradığınızı bildirin
                Serial.print("Calling to : ");
                Serial.println(remoteNumber);
                Serial.println();

                // "remoteNumber" ara
                remoteNumber.toCharArray(charbuffer, 20);

                if (vcs.voiceCall(charbuffer)) {
                    Serial.println("Call Established. Enter line to end");
                    // Kullanıcıdan input bekler bu satırda
                    while (Serial.read() != '\n' && (vcs.getvoiceCallStatus() ==
TALKING));
                    // Ve arama biter
                    vcs.hangCall();
                }
                Serial.println("Call Finished");
                remoteNumber = "";
                Serial.println("Enter phone number to call.");
            } else {
                Serial.println("That's too long for a phone number. I'm
forgetting it");
                remoteNumber = "";
            }
        } else {
            // Mesaja göndermek için son karakteri ekler
            if (inChar != '\r') {
                remoteNumber += inChar;
            }
        }
    }
}
```

```
}  
}
```

## Seri Port Çıktısı



The screenshot shows a serial port monitor window titled "COM4 (Arduino/Genuino Uno)". The window contains the following text:

```
Make Voice Call  
GSM initialized. → GSM BAŞLATILDI  
Enter phone number to call. → ARAYACAĞINIZ NUMARAYI GIRECEKSİNİZ  
Calling to : 053212345675  
  
Call Finished  
Enter phone number to call. → ARAYACAĞINIZ YENİ NUMARAYI GIRMENİZİ BEKLİYOR
```

The window also features a "Gönder" button at the top right, a "Otomatik Kaydırma" checkbox at the bottom left, and dropdown menus for "NL ve CR ile birlikte.", "9600 baud", and a "Clear output" button at the bottom right.

## Arduino Üzerinden GSM2 Click ile SMS gönderme

Aşağıdaki tabloda verilen Arduino kodlarını kopyalayıp Arduino Editörüne yapıştırıyoruz ve kartımıza yüklüyoruz. Daha sonra **Serial Port Ekranını** açıyoruz. Açtığımız anda kodumuz çalışmaya başlıyor. **Seri port ekranında Baud Rate 'in kodumuzda yazdığımızla aynı olmasına dikkat ediniz. Aksi takdirde çıktı alamazsınız.**



```
#include <GSM.h>
// PIN Numarası. Eğer pin numarası varsa burada tanımlayın
#define PINNUMBER ""

// Kütüphane örneğini başlat
GSM gsmAccess;
GSM_SMS sms;

void setup() {
  // Seri haberleşmenin başlaması seri porta bağlanmayı beklenmesi
  Serial.begin(9600);
  while (!Serial) {
    ; // seri porta bağlanmayı beklenmesi. Sadece yerel USB bağlantı
    noktası için geçerli
  }

  Serial.println("SMS Messages Sender");

  // Bağlantı parametresi
  boolean notConnected = true;

  // GSM shield başlat
  // // Eğer PIN kodunuz varsa "begin()" parametresi içine "PINNUMBER"
  // sabitini tanımlıyoruz
  while (notConnected) {
    if (gsmAccess.begin(PINNUMBER) == GSM_READY) {
      notConnected = false;
    } else {
      Serial.println("Not connected");
      delay(1000);
    }
  }

  Serial.println("GSM initialized");
}

void loop() {

  Serial.print("Enter a mobile number: ");
  char remoteNum[20]; // SMS göndereceğin telefon numarası
  readSerial(remoteNum);
```

```
Serial.println(remoteNum);

// SMS Metni
Serial.print("Now, enter SMS content: ");
char txtMsg[200];
readSerial(txtMsg);
Serial.println("SENDING");
Serial.println();
Serial.println("Message:");
Serial.println(txtMsg);

// SMS gönder
sms.beginSMS(remoteNum);
sms.print(txtMsg);
sms.endSMS();
Serial.println("\nCOMPLETE!\n");
}
/*
Serial'dan gelen input
*/
int readSerial(char result[]) {
    int i = 0;
    while (1) {
        while (Serial.available() > 0) {
            char inChar = Serial.read();
            if (inChar == '\n') {
                result[i] = '\0';
                Serial.flush();
                return 0;
            }
            if (inChar != '\r') {
                result[i] = inChar;
                i++;
            }
        }
    }
}
```

,

