**BlueToothe AT-09**

**רקע כללי:**

רכיב הבלוטוס מתקשר באמצעות RF, הוא מתקשר בתחום התדרים של 2.4GHz. ישנם מספר דגמים וסוגים לרכיב זה, הרכיב שמדובר בסיכום זה הוא רכיב שמורכב על לוח מתאם, והוא נקרא AT-09. ממנו יוצאים 6 פינים כפי שיתואר כאן:

1. **VCC**: הרכיב יכול לקבל מתח הפעלה בין 3.6V ל- 6V
2. **GND**: אדמה
3. **RX**: תקשורת סיראלית UART, זהו צד הרסיבר
4. **TX**: תקשורת סיראלית UART, זה צד המשדר
5. **EN**: שליטה על כיבוי או הפעלה לצורך מצב AT Command, על זה נסביר בהרחבה בהמשך
6. **STATUS**: מצב המכשיר, פנוי/עסוק

ה-Shiled של רכיב ה-Bluetooth מבוסס על רכיב cc2541f256TI ששייך לחברת Texas Instremants.

רכיב זה יכול לשדר על מספר עוצמות שונות של dBm, כאשר זה נע בין 0 ל 94- שככול שנתקב יותר ל-0 צריכת הזרם תעלה ותגיע במקסימום ל-18.2mA במצב TX.

רכיב זה יכול לשדר עד 1Mbps במצב של -94dBm. בנוסף יש לו טיימר שינה, פסיקות חיצוניות ועוד דברים שפחות רלוונטים בשל ה-Shiled שחוסך לנו התעסקות מיותר בפרטים הקטנים.

חיווט הרכיב:

הרכיב יכול לעבוד בעזרת ההדקים TX, RX, VCC, GND, שאר ההדקים הם אופציה בלבד. מה זה אומר, אם אנו רוצים לתקשר בשני כיוונים, כלומר גם שהרכיב ישלח מידע וגם יקבל מידע, אנו נצטרך להשתמש בהדק ה-STATUS, כדי לדעת את המצב שלו, למשל, אם הוא כרגע מקבל מידע, הוא לא יכול לשלוח מידע, וכך נוכל להפעיל את התוכנית שלנו בצורה טובה ויעילה מבלי איבוד מידע

ההדק של ה-EN הוא שמיש רק אם אנו מעוניינים לעבור למצב של פקודות AT Commands בזמן שהתוכנית שלנו רצה. העברת ההדק ל-'1' לוגי, יכבה אותו, העברה שוב ל-'0' יפעיל אותו, ואז אנחנו שוב ניהיה במצב של פקודות.

הרכיב עובדת על רמת מתח של 3.3V כלומר, אופן התקשרות שהרכיב מוציא ומקבל בהדקים של ה-RX וה- TX הוא 3.3V. ישנם בקרים של 5V שיכולים לתקשר עם הרכיב הזה אם מצב ה-'1' לוגי שלהם נמוך מ-3.3V. בקר של Atmel328P, יכול לתקשר בצורה זו.

מומלץ ביציאה של הבקר שלנו, כלומר TX לחבר שני נגדים של 1K ו- 2K אום, בכדי ליצור מפצל מתח בשביל לא לפגוע בבקר של ה-BlueTooth, אך דבר זה אינו חובה.

**אופן הפעלה:**

בדומה להרבה רכיבים אחרים, גם רכיב זה מתקשר בעזרת פקודות AT Command, רכיב זה מחזיק פקודה שנקראת HELP דרכה ניתן להשיג את כל המידע על הפקודות שקיימות במערכת.

להלן הפקודות:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Command Description

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* AT+BAUD Get/Set baud rate

\* AT+RESET Software reboot

\* AT+ROLE Get/Set current role.

\* AT+DISC Disconnect connection

\* AT+ADVEN Broadcast switch

\* AT+ADVI Broadcast interval

\* AT+NINTERVAL Connection interval

\* AT+POWE Get/Set RF transmit power

\* AT+NAME Get/Set local device name

\* AT+LADDR Get local bluetooth address

\* AT+VERSION Get firmware, bluetooth, HCI and LMP version

\* AT+TYPE Binding and pairing settings

\* AT+PIN Get/Set pin code for pairing

\* AT+UUID Get/Set *system* SERVER\_UUID .

\* AT+CHAR Get/Set *system* CHAR\_UUID .

\* AT+INQ Search from device

\* AT+RSLV Read the scan *list* MAC address

\* AT+CONN Connected scan *list* device

\* AT+CONA Connection specified MAC

\* AT+BAND Binding from device

\* AT+CLRBAND Cancel binding

\* AT+GETDCN Number of scanned *list* devices

\* AT+SLEEP Sleep mode

\* AT+HELP List all the commands

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

כמו שהסברתי קודם, המערכת מתקשרת בעזרת תקשורת סיראלית, כלומר UART.

המערכת מגיעה בקצב של 9600 כ-Default. ניתן לשנות את ערך זה על ידי הפקודות. דבר שחשוב לציין, למערכת יש זיכרון בלתי מחיק, שבו מאוחסן כל שינוי שנבצע, כלומר אין צורך לטעון מחדש את כל הערכים שאנו מעוניים לקבוע למערכת כל פעם בעת חיבור למתח, אלא מספיק הגדרה חד פעמית של המערכת ונתנוים ישמרו.

מכל הפקודות שקיימות, אנו זקוקים רק לכמה בודדות בשביל להפעיל את המערכת, אני אעבור על רוב הפקודות שקיימות עם הסברים ודוגמאות של אופן ביצוע הפקודה.

**פקודת AT+BAUD:**

פקודה זו קובעת את קצב העברת מידע בין הבקר לבין המערכת, כאשר זה נקבע לפי ערכים שבין 0-4. 4=9600, 3=19200, 2=38400, 1=57600, 0=115200

אופן רישום הפקודה הוא: AT+BAUD4

בתגובה אנו נקבל: +BAUD=9600 ,ושורה מתחת נקבל OK.

**פקודת AT+RESET:**

פקודה זו תחזיר את המערכת למצב הדיפולטיבי שלה, כלומר, תאפס את הסיסמא, תחזיר את שם הרכיב למקור, תאפס את קצב המידע וכו.

אופן רישום הפקודה היא: AT+RESET לאחריה אנו נקבל כרגיל OK.

**פקודת AT+ROLE:**

פקודה זו למעשה קובעת מי מהרכבים יהיה מאסטר ומי יהיה SLAVE, כאשר המערכת מגיעה במצב של SLAVE.

הערך 0 הוא SLAVE, והערך 1 הוא MASTER.

אופן רישום הפקודה הוא AT+ROLE1, לאחר מכן אנו נקבל +ROLE=1 וכמובן OK בשורה מתחת

חשוב לציין, רכיב שהוגדר כ-MASTER, לא יהיה ניתן לזהות אותו כאשר תחפשו רכיבים, בחיפוש רכיבים ניתן לאתר רק רכיבים שמוגדרים כ-SLAVE, ה-MASTER הוא זה שיוצר את החיבור של הרכיב.

**פקודת AT+NAME:**

בעזרת פקודה זו אנו יכולים לשנות את השם של הרכיב, השם שלו מגיע בדרך כלל כ-"?".

אופן רישום הפקודה הוא AT+NAME**MYNAME**, לאחר מכן אנו נקבל את התגובה של המערכת כ- +NAME=MYNAME וכמובן שורה מתחת OK

**פקודת AT+LADDR:**

פקודה זו תתן לנו את כתובת ה-MAC של הרכיב שלנו. אופן שליחת הפקודה הוא פשוט כמו שהיא. AT+LADDR, בתגובה אנו נקבל +LADDR=01:02:03:04:1B:29כמובן שהכתובת משתנה מרכיב לרכיב, בעזרת כתובת זו, אנו נוכל להתחבר לרכיב שאליו אנו מעוניינים להתחבר. כמובן שנצטרך לדעת אותה מראש בשביל להתחבר אליו.

חשוב לציין, הכתובת של הרכיב לא משתנה מהפעלה להפעלה של הרכיב. בנוסף לא ניתן לשנות את הכתובת של הרכיב בעזרת פקודות ה-AT Commands.

**פקודת AT+VERSION:**

פקודה זו תתן לנו את הגרסה של הרכיב שלנו, ניתן לעדכן את הגירסה של הרכיב, אך תהליך זה מורכב ומיותר.

אופן כתיבת הפקודה היא בדיוק כמו שהיא. AT+VERSION בתגובה אנו נקבל בחזרה מהרכיב את הגרסה, במקרה שלי זה JDY-09-V4.3

**פקודת AT+PIN:**

כאשר נכתוב את הפקודה כמו שהיא, אנו נקבל את הסיסמא הנוחכית. אם אנו מעוניינים לשנות את הסיסמא, אנו נכתוב AT+PIN123456 ואנו נקבל בחזרה מהמערכת את הסיסמא החדשה, +PIN=123456 חשוב לציין שהסיסמא חייבת להיות לפחות בת 6 תווים.

אגב, הסיסמא המקורית של הרכיב היא 123456

כאשר נרצה לראות את הסיסמא, אנו נרשום AT+PIN ואנו נקבל חזרה מהמערכת את הסיסמא: +PIN=123456

**פקודת AT+CONA:**

פקודה זו מחברת בין שני רכיבי Bluetooth בהתבסס על כתובת ה-MAC שלהם שאותה ניתן למצוא בעזרת הפקודה AT+LADDR שעליה הסברנו קודם

אם אנו מעוניינם להתחבר לרכיב שאנו יודעים את הכתובת שלו, למשל הכתובת של הרכיב שאנו מעוניינם היא: 01:02:03:04:1B:29

אנו נרשום: AT+CONA010203041B29 כלומר את הכתובת רק ללא ה ":" שמפרידים בין חלקי הכתובת.

בתגובה אנו נקבל +Conncted ושורה מתחת את ה-OK

**פקודת AT+TYPE:**

פקודה זו קובעת את אופן ההתחברות של הרכיב, כלומר להתחבר עם סיסמא או בלי סיסמא. כאשר TYPE=0 זה ללא סיסמא, כאשר TYPE=1 זה עם סיסמא.

הרכיב מגיע במצב של 0. אופן כתיבת הפקודה הוא AT+TYPE0 ואז אנו נקבל מהמערכת בחזרה TYPE=0+ וכמובן שורה מתחת OK.

**פקודת AT+SLEEP:**

פקודה זו תשלח את הרכיב שלנו לישון כלומר הרכיב שלנו יעבור למצב הרבה יותר חסכוני מבחינת צריכת זרם, ניתן יהיה להעיר אותו בעזרת הפקודה AT, יקח לו חצי שניה לחזור למצב עבודה תקין.

אופן שליחת הפקודה היא AT+SLEEP בתגובה אנחנו נקבל +SLEEP:OK וזה למעשה אומר שהמערכת שלנו עברה למצב שינה.

**שלבי הפעלה:**

קודם כל נתחיל מחיבור הרכיב למערכת, ניתן לחבר אותו לרכיב USB שממיר לתקשורת סיראלית רק בשביל לוודא שהוא אכן מתקשר.

פקודת AT אמורה להחזיר OK, למעשה זה קיצור של המילה ATTENTION.

אנו נגדיר את הרכיב כ-SLAVE, נמצא מה הכתובת MAC שלו, ובנוסף נשנה לו שם לשם שאנו רוצים שיקרא הרכיב.

רכיב נוסף נגדיר כ-MASTER, נשנה גם לו את השם לשם שאנו מעוניינים. כעת כל מה שנשאר לעשות זה לחבר בין שני הרכיבים.

נשתמש בפקודת AT+CONA כמובן שנשתמש בכתובת הרכיב של ה-SLAVE שמצאנו קודם. אחרי שאנחנו נקבל CONNECTED אנו יודעים שהרכיבים מחוברים. עכשיו, כל מה שנרשום לרכיב הוא ישלח את זה לרכיב השני.

עכשיו הרכיב לא מתייחס לפקודות, וכל מה ששולחים לו הוא שולח, למשל AT, הוא ישלח AT לרכיב השני.

חשוב לציין שהתקשורת היא דו כיוונית, כלומר ה-MASTER יכול לשלוח ל-SLAVE וכנל הפוך. אין צורך להגדיר או לשנות כלום, פשוט לשלוח תווים דרך התקשורת הסיראלית.

ברגע שהרכיבים יהיו מחוברים, הנורה תעבור ממצב של הבהוב למצב של דלוקה באופן קבוע!

הערה: שים לב שבעת חיבור הרכיב הוא מהמהב בקצב של שניה, כלומר שניה דלוק ושניה כבוי. זה אומר שהרכיב נמצא במצב של SLAVE.

אם ההבהוב מהיר יותר, בקצב של חצי שניה, הרכיב לא נמצא במצב שאפשר להתחבר דרכו או אליו. הסיבה לכך, מכיוון שהרכיב מוגדר כרגע כ-MASTER, ורק הוא יכול להתחבר לרכיבים אחרים.

בשביל שהרכיבים יוכלו להתחבר, חייב שרכיבי ה-SLAVE יהיו במצב ללא סיסמא. אל דאגה, עדיין תתבקש סיסמא לכל רכיב חיצוני אחר שיבקש להתחבר אל הרכיב. כלומר עלינו לוודא שהרכיב אכן נמצא במצב של TYPE=0. הרכיב מגיע כך ב-DEFAULT שלו.

דוגמא לתוכנית של רכיב MASTER

int main(void)

{

bluetooth test;

test.setDeviceAsMaster();

test.setDeviceName("MyDevice");

test.connectToDevice("010203041ADE");

char arr[10] = "123456789";

while (1)

{

test.sendData(arr);

test.sendData("1234567");

arr[0]++;

}

}

דוגמא של תוכנית כרכיב SLAVE:

int main(void)

{

bluetooth test;

char MAC[20];

test.setDeviceAsMaster();

test.setDeviceName("MyDevice2");

test.getDeviceMACaddress(MAC);

char arr[20] = "123456789";

while (1)

{

//the function return 1 if new data is received

if (test.getData(arr))

{

test.UARTstring("We received DATA!");

test.UARTstring(arr);

}

}

}