DAFTAR GAMBAR

DAFTAR TABEL

LAMPIRAN

1 . Data Sheet NodeMCU Esp 8266

1 Introduction

Espressif Systems' Smart Connectivity Platform (ESCP) of high performance wireless SOCs, for mobile platform designers, provides unsurpassed ability to embed Wi-Fi capabilities within other systems, at the lowest cost with the greatest functionality.

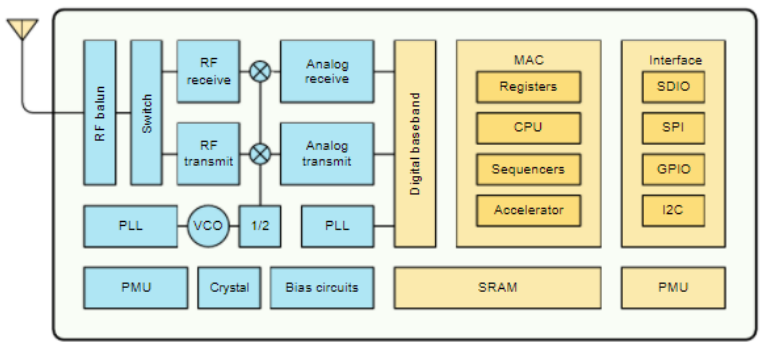
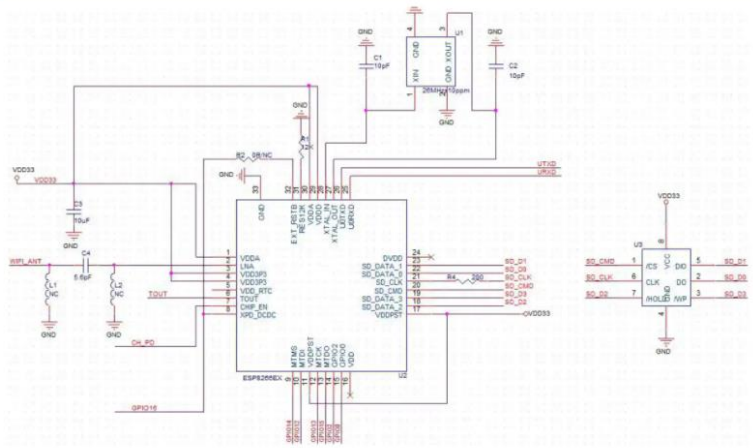


Figure 1: ESP8266 Block Diagram

Features

- 802.11 b/g/n protocol
- Wi-Fi Direct (P2P), soft-AP
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLL, regulators, and power management units
- +19.5dBm output power in 802.11b mode
- Integrated temperature sensor
- Supports antenna diversity
- Power down leakage current of < 10uA
- Integrated low power 32-bit CPU could be used as application processor
- SDIO 2.0, SPI, UART
- STBC, 1×1 MIMO, 2×1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4μs guard interval
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)

4 Application Diagram



7 Specifications

7.1 Current Consumption

The following current consumption is based on 3.3V supply, and 25°C ambient, using internal regulators. Measurements are done at antenna port without SAW filter. All the transmitter's measurements are based on 90% duty cycle, continuous transmit mode.

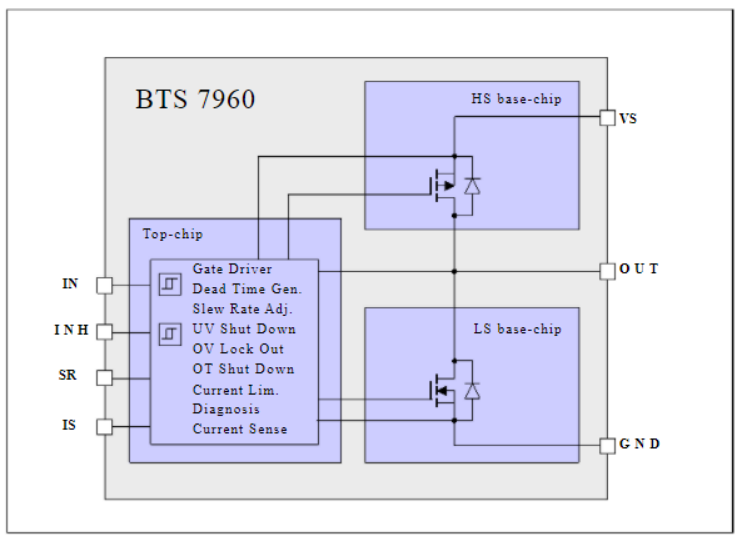
Mode	Min	Typ	Max	Unit
Transmit 802.11b, CCK 1Mbps, P_{OUT}=+19.5dBm		215		mA
Transmit 802.11b, CCK 11Mbps, P_{OUT}=+18.5dBm		197		mA
Transmit 802.11g, OFDM 54Mbps, P_{OUT} =+16dBm		145		mA
Transmit 802.11n, MCS7, P_{OUT}=+14dBm		135		mA
Receive 802.11b, packet length=1024 byte, -80dBm		60		mA
Receive 802.11g, packet length=1024 byte, -70dBm		60		mA
Receive 802.11n, packet length=1024 byte, -65dBm		62		mA
Standby		0.9		mA
Deep sleep		10		uA
Power save mode DTIM 1		1.2		mA
Power save mode DTIM 3		0.86		mA
Total shutdown		0.5		uA

7.2 RF Performance

The following are measured under room temperature conditions with 3.3V and 1.1V power supplies.

Description	Min	Typical	Max	Unit
Input frequency	2412		2484	MHz
Input impedance		50		Ω
Input reflection			-10	dB
Output power of PA for 72.2Mbps	14	15	16	dBm
Output power of PA for 11b mode	17.5	18.5	19.5	dBm
Sensitivity				
CCK, 1Mbps		-98		dBm
CCK, 11Mbps		-91		dBm
6Mbps (1/2 BPSK)		-93		dBm
54Mbps (3/4 64-QAM)		-75		dBm
HT20, MCS7 (65Mbps, 72.2Mbps)		-71		dBm
Adjacent Channel Rejection				
OFDM, 6Mbps		37		dB
OFDM, 54Mbps		21		dB
HT20, MCS0		37		dB
HT20, MCS7		20		dB

2 . Data Sheet Motor Driver BTS 7960



2 Pin Configuration

2.1 Pin Assignment

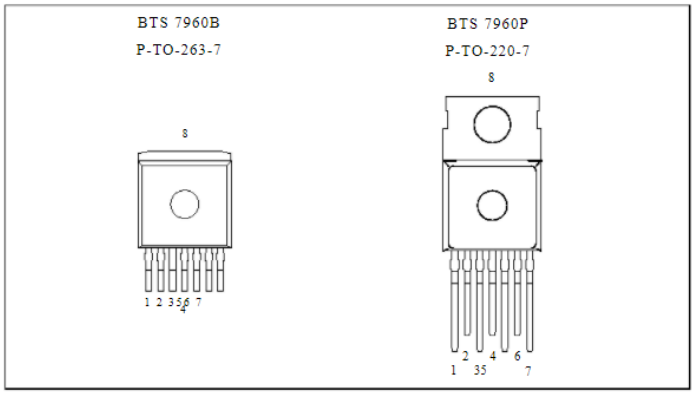


Figure 3 **Pin Assignment BTS 7960B and BTS 7960P (top view)**

2.2 Pin Definitions and Functions

Pin	Symbol	I/O	Function
1 G N D		-	**Ground**
2	IN	I	Input Defines whether high- or lowside switch is activated
3 I N H		I	Inhibit When set to low device goes in sleep mode
4,8	**O U T**	**O**	**Power output of the bridge**
5 S R		I	Slew Rate The slew rate of the power switches can be adjusted by connecting a resistor between SR and GND
6	IS	O	Current Sense and Diagnosis
7	**VS**	-	**Supply**

3 Maximum Ratings

-40 °C < T_j < 150 °C (unless otherwise specified)

Pos	Parameter	Symbol	Limits		Unit	Test Condition
			min	max		
Electrical Maximum Ratings						
3.0.1	Supply voltage	V_{VS}	-0.3	45	V	
3.0.2	Logic Input Voltage	V_{IN} VINH	-0.3	5.3	V	
3.0.3	HS/LS continuous drain current	ID(HS) ID(LS)	-40	40 1)	A	TC < 85°C switch active
3.0.4	HS pulsed drain current	ID(HS)	-60	60 1)	A	TC < 85°C tpulse = 10ms
3.0.5	LS pulsed drain current	ID(LS)	-60	60 1)	A	
3.0.6	Voltage at SR pin	V_{SR}	-0.3	1.0	V	
3.0.7	Voltage between VS and IS pin	VVS - VIS	-0.3	45	V	
3.0.8	Voltage at IS pin	V_{IS}	-20	45	V	
Thermal Maximum Ratings						
3.0.9	Junction temperature	T_j	-40	150	°C	
3.0.10	Storage temperature	T_{stg}	-55	150	°C	
ESD Susceptibility						
3.0.11	ESD susceptibility HBM	V_{ESD}			kV	according to EIA/ JESD 22-A 114B
	IN, INH, SR, IS		-2	2		
	OUT, GND, VS		-6	6		

1) Maximum reachable current may be smaller depending on current limitation level

```
//SOURCE CODE ALAT FOGGING H2O2 TUGAS
//TUGAS AKHIR UNTAG 2021
//PROGRAM STUDI ELKA
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include
<BlynkSimpleEsp8266.h>
//Motor PINs
#define IN1 D1 //BTS MOTOR 1
R_PWM
#define IN2 D2 //BTS MOTOR 1
L_PWM
#define IN3 D3 //BTS MOTOR 2
R_PWM
#define IN4 D4 //BTS MOTOR 2
L_PWM
#define TRIG D8
#define ECHO D7
bool maju = 0;
bool kiri = 0;
bool kanan = 0;
bool belakang = 0;
int Speed;

char auth[] = "CrNadleXMTyOXDWT_Xht9vHAC36tLzBl";
//Enter your Blynk application auth token
char ssid[] = "dhoni"; //Enter
your WIFI name
char pass[] = "123456789";
//Enteryour WIFI passowrd

WidgetLCD lcd(V5);
```

```
                void setup() {
                  Serial.begin(9600);

                  pinMode(IN1, OUTPUT);
                  pinMode(IN2, OUTPUT);
                  pinMode(IN3, OUTPUT);
                  pinMode(IN4, OUTPUT);
                  pinMode(TRIG, OUTPUT);
                  pinMode(ECHO,
                INPUT_PULLUP);

                  void mobil_berhenti();

                  Blynk.begin(auth, ssid, pass);}

                //code untuk membuat parameter pada blynk dan
                dihubungkan ke nodemcu
                BLYNK_WRITE(V0) {maju =
                param.asInt();}

                BLYNK_WRITE(V1) {kiri =
                param.asInt();}
                BLYNK_WRITE(V2) {kanan =
                param.asInt();}
                BLYNK_WRITE(V3) {belakang
                = param.asInt();}

                //fungsi motor
                void smartcar() {if (maju == 1)
                {mobilmaju();
                   Serial.println("mobil maju");}
                  else if (kiri == 1)
                {mobilb_kiri();
  Serial.println("belok kiri");  }
```

```
else if (kanan == 1)
{mobilb_kanan();
   Serial.println("belok kanan");}
  else if (belakang == 1)
{mobilmundur();
   Serial.println("mundur");}
  else if (maju == 0 && kiri == 0
&& kanan == 0 )
{mobil_berhenti();
   Serial.println("Berhenti");}

//proses
 void loop(){
 Blynk.run();
 lcd.clear(); //Use it to clear the
LCD Widget
 smartcar();

  //Coding Level Air
  digitalWrite(TRIG, LOW);
  delayMicroseconds(2);

  digitalWrite(TRIG, HIGH);
  delayMicroseconds(20);

  digitalWrite(TRIG, LOW);
  int jarak =
pulseIn(ECHO,HIGH,26000);

  jarak = jarak/58;

  Serial.print("Tinggi Air : ");
  Serial.print(jarak);
  Serial.println("cm");
```

```
lcd.print(0,0, "JARAK : ");
  lcd.print(7,0, jarak);
  Blynk.virtualWrite(V4, jarak);
  delay(500);

if(jarak>40){Blynk.notify("TAN
GKI KOSONG! SEGERA ISI
ULANG LIQUID H2O2!!");}
//fungsi gerakan motor
void mobilmaju()
{digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);}
void mobilb_kiri()
{digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);}
void mobilb_kanan()
{digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);}
void mobil_berhenti()
{digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);}
void mobil_mundur()
{digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);}
```

3. Source Code