Default baud rate of LED Display is 115200	
Except for codes for 0xd1-0xd9, 0xa0-ab and 0xf0-0xf6 all other codes only change the display memory, thus you have to excute the "0xd1" code (Function of refresh the display) to	Wrong Example: Write, SVZ Sing/3, 17, positive, "RPM"); Write, Sk16, Pattern(f. 45, positive, 0); Draw, Rectangle(0, 0, 127, 127, positive); //Without excute the Show_Display_Momeny() function the change only in the memory, it won't display */
display the changes in the display memory after you write a character.	Correct Example: Write, SAY, String(7, 17, positive, "RPM"); Write, BAY String(7, 14, positive, 0); Write, BAY (6, Pattern(1, 45, positive, 0);

	string , pattern or draw a line.		Write_8X16_Pattern(1, 45, positive, 0); Draw_Rectangle(0, 0, 127, 127, positive); Show_Display_Momery();
			/"With the execution of Show_Display_Memory() fucntion, the change of display memory will be displayed"/
Code	Function	Sequence of HEX command mode through UART	API for Arduino
N/A	Sent a image(192X64 bitmap) to LED (An array consist of 1536 bytes bitmap information)	A "for" loop to send 1536 bytes user define display information Wait until receive a module available byte (E) from LED 3. Wait 2ms	for (= 0 ; < 158; +++ Serial.write(User_define_array(i)); while (Serial.read() !=E') () delay(2);
0x80	Write a SX7 Character	1. Send ob/60 2. Send which line to put this character 3. Send which doumn to put this character 4. Send character 4. Self code 5. Wait until receive a module available byte (°E') from LED 6. Wait 2ms	void Wirrie SX7_Character(int line, int column, int negative, char Char) { Serial write(Role); Serial write(Role); Serial write(column); Serial print(Char); while (Serial read) !=E' () delay(2); }
0x81	Write a 8X8 String	Send which line to start the string Send which line to start the string Send which doumn to start the string Send string	vaid Wire, SXT_String(int line, int column, int negative, char * string) { Serial write(SM); Serial write(SM); Serial write(Ine); Serial write(column); Serial print(string); while (Serial.radd) I=E) () delay(2); }
0x82	Write a 8X16 Character	1. Send olv60 2. Send which line to put this character 3. Send which downn to put this character 4. Send character 4. Send character 4. Send character 4. Send character 4. Self clode 5. Walt until receive a module available byte(E) from LED 6. Walt 2ms	void Write, BX16. Character(int line, int column, int negative, char Char) { Serial write(Barg); Serial write(Barg); Serial write(Char); Serial write(Column); Serial print(Char); white (Serial.read) !=E) {} delay(2); }
0x83	Write a 8X16 String	Send which line to starty the string Send which line to starty the string Send which down to start the string Send string Walt until receive a module available byte(E) from LED S. Walt 2ms	void Wirle EX16. String(int line, int column, int negative, char * string) (Serial write(Das); Serial write(Das); Serial write(Das); Serial write(Dunn); Serial print(string); while (Serial.nad) (I=E) () delay(2); }
0x84	Dsiplay a 8X8 pattern	Send but Justit X coordinate of pattern Send the Up Left X coordinate of pattern Send the Up Left Y coordinate of pattern Send the 10 of pattern Switz wall receive a module available byte (E) from LED 6. Wait 2ms	void Viric BX2, Pattern(int Up_Left_Xpos, int Up_Left_Ypos, int negative, int Pattern_ID) { Serial winte(Up_Left_Xpos); Serial winte(Up_Left_Xpos); Serial winte(Up_Left_Xpos); Serial winte(Pattern_ID); winte(Pattern_ID); winte(Serial.read() I=F) () delay(2); }
0x85	Dsiplay a 8X16 pattern	Send to Up Left X coordinate of pattern Send the Up Left X coordinate of pattern Send the Up Left Y coordinate of pattern Send the 10 For pattern Swall the 10 For pattern	void Write, 8X16, Patternt int Up_Left_Xpos, int Up_Left_Ypos, int negative, int Pattern_ID) { Serial write(DoS); Serial write(Up_Left_Xpos); Serial write(Up_Left_Xpos); Serial write(Up_Left_Ypos); Serial write(Up_Left_Ypos); Serial write(Up_Left_Xpos); Write(Serial raed() I=E) { delay(2); } delay(2); }
0x86	Dsiplay a 16X16 pattern	1. Send oibe8 2. Send the Up Left X coordinate of pattern 3. Send the Up Left Y coordinate of pattern 4. Send the Up Contained of pattern 4. Send the 10 of pattern 5. Wait until receive a module available byte (E) from LED 6. Wait 2ms	void Winc. 16/16, Pattern (int Up_Left_Xpos, int Up_Left_Ypos, int negative, int Pattern_ID) { Serial wint(0,0)s(): Serial wint(0,0) Left_Xpos); Serial wint(0,0 Left_Xpo
0x87	Dsiplay a 32X32 pattern	Send to Use! Send the Up Left X coordinate of pattern Send the Up Left X coordinate of pattern Send the Up Left X coordinate of pattern Send the Do of pattern Swall the ID of pattern Swall tri	soid Wine, 32/32, Pattern (int Up_Left_Xpos, int Up_Left_Ypos, int negative, int Pattern_ID) { Serial wine(Do, Left_Xpos); Serial wine(Do, Left_Xpos); Serial wine(Do, Left_Ypos); Serial wine(Do, Left_Ypos); Serial wine(Pattern_ID); wine(Serial raed() I=E) { delay(2); } }
0x90	Draw a line	1. Send 0:60 2. Send he X coordinate of first point 3. Send he Y coordinate of first point 3. Send he Y coordinate of second point 5. Send the Y coordinate of second point 5. Send the Y coordinate of second point 6. Send of roof local folias produce (1 to positive, 0 for negative) 7. Wait until receive a module available byte (E) from LED 8. Wait 2ms	said Draw Line(int XD_Pos, int Y0_Pos, int X1_Pos, int Y1_Pos, int negative) { Serial wire(XD_Pos); Serial wire(XD_Pos); Serial wire(XP_Pos); Serial wire(
0x91	Draw a Rectangle	1. Send old: 2. Send the X coordinate of up left corner 3. Send the Y coordinate of up left corner 4. Send the Y coordinate of bottom right corner 5. Send the Y coordinate of bottom right corner 5. Send the Y coordinate of bottom right corner 7. Send the Y coordinate of bottom right corner 8. Send the Y coordinate of bottom right corner 9. Send the Y coordinate o	void Draw, Rectandgel, eft X0_Pos, int Y0_Pos, int X1_Pos, int Y1_Pos, int negative) { Serial write(D0, Pos); Serial write(X0, Pos); Serial write(X1_Pos); Serial write(X1_Pos); Serial write(X1_Pos); Serial write(X1_Pos);
0x92	Draw a filled Rectangle	1. Send to Mo2 2. Send the X-coordinate of up left corner 3. Send the Y-coordinate of tup left corner 4. Send the Y-coordinate of bottom right corner 5. Send the Y-coordinate of bottom right corner 5. Send the Y-coordinate of bottom right corner 6. Send of to Of to Galpay mode (1 for possite, 0 for negative) 7. Walt until receive a module available byte (*E) from LED 8. Wait 2ms	void Draw_Filled_Rectangle(int X0_Pos, int Y0_Pos, int X1_Pos, int Y1_Pos, int negative) { Serial write(X0_Pos); Serial write(X0_Pos); Serial write(X1_Pos); Serial write(X1_Po
0x93	Draw a Square	Send 0x3 Send the X coordinate of up left corner Send the Y coordinate of up left corner Send the Y coordinate of up left corner Send to Post or display mode (1 for positive, 0 for negative) Send 1 or 10 for display mode (1 for positive, 0 for negative) Send to 10 for display mode (1 for positive, 0 for negative) Twat until receive a module available byte (E) from LED Twat 2ms	void Draw_Square(int X0_Pos, int Y0_Pos, int width, int negative) { Serial write(DAS); Serial write(X0_Pos); Serial write(X0_Pos); Serial write(Y0_Pos); Serial write(Y0_Pos)
0x94	Drawa Circle	Send 0x84 Send the X coordinate of the center Send the X coordinate of the center Send the Y coordinate of the center Send to Provide the radius of this critical Send to 10 for display mode (1 for positive, 0 for negative) SN'ast until receive a module available byte (E) from LED T. Wat 2ms	void Draw_Circle(int X0_Pos, int Y0_Pos, int radius, int negative) { Serial write(DA4); Serial write(X0_Pos); Serial write(X0_Pos); Serial write(Y0_Pos); Serial write(Y0_Pos
0x95	Draw a filled Circle	Send the X-coordinate of the center Send the Y-coordinate of the center Send the Y-coordinate of the center Send the radius of this circle Send the radius of the receive a module sivaliable byte (E) from LED T. Wait 2ms.	void Draw_Filled_Circle(int X0_Pos, int Y0_Pos, int radius, int negative) { Serial write(N0_Pos); Serial write(X0_Pos); Serial wr

0x96	Draw a tip upward Triangle	1. Send to 066 2. Send the X-coordinate of the tip 3. Send the Y-coordinate of the tip 3. Send the Y-coordinate of the tip 4. Send the height of the tip to the bottom 5. Send 1 or 0 for display mode (1 for positive, 0 for negative) 6. Wait until receive a module available byte (E) from LED 7. Wait 2ms	Vaid Draw, Triangle, Up, Ward(int X0_Pos, int Y0_Pos, int height, int negative) { Serial wire(D0P), Pos ; Serial wire(P0P, Pos ;); while (Serial read) !=E? () }
0x97	Draw a filled tip upward Triangle	1. Send to Mo? 2. Send the X coordinate of the tip 3. Send the Y coordinate of the tip 4. Send the height of the tip to the bottom 5. Send 1 or 0 for display mode (1 for positive, 0 for negative) 6. Send 1 or 0 for display mode (1 for positive, 0 for negative) 7. Wast 2ms The Mode of the Year of Yea	sid Draw, Filled_Triangle_Up_Ward(int X0_Pos, int Y0_Pos, int height, int negative) { Serial write(DX)_Pos); Serial write(PX)_Pos); Serial write(PX)_Pos); Serial write(PX)_Pos); Serial write(PX)_Pos); Serial write(PX)_Pos); Ward(BX)_Pos); Ward(
0x98	Draw a tip downward Triangle	1. Send to 0:08 2. Send the X-coordinate of the tip 3. Send the Y-coordinate of the tip 3. Send the Y-coordinate of the tip 5. Send the height of the tip to the top 5. Send 1 or 0 for display mode (1 for positive, 0 for negative) 5. Send 1 or 0 for display mode (1 for positive, 0 for negative) 6. Wait unit receive a module available byte (E') from LED 7. Wait Zms	Laid Draw, Triangle, Down, Ward(int X0_Pos, int Y0_Pos, int height, int negative) (Serial wire(X0_Pos); Wire(Serial road) (1-E) () delay(Z);
0x99	Draw a filled tip downward Triangle	Send 0x99 Send he X coordinate of the tip Send he Y coordinate of the tip Send he Y coordinate of the tip Send the Y coordinate of the tip Send to Y coordinate of the tip Send to Y coordinate of the tip Send to Y coordinate of the Y	J und Draw_Filled_Triangle_Down_Ward(int X0_Pos, int Y0_Pos, int height, int negative) { Serial wint(X0,09); Serial wint(X0_Pos); Serial wint(X0_Pos); Serial wint(X0_Pos); Serial wint(X0_Pos); Serial wint(X0_Pos); Serial wint(X0_Pos); While (Serial read) I=E) () delay(Z);
0x9a	Draw a 5p leftward Triangle	Send 0x9a Send the X coordinate of the tip Send the Y coordinate of the tip Send the Y coordinate of the tip Send the Y coordinate of the tip Send to Y coordinate of the tip Send to Y coordinate of the tip Send to Y coordinate of the	J void Draw_Triangle_Left_Ward(int X0_Pos, int Y0_Pos, int width, int negative) { Serial winte(X0e); Serial winte(X0,Pos); Serial winte(X0,Pos); Serial winte(x0,Pos); Serial winte(x0,Y0); Serial winte(x0,Y0); While (Serial.read) I=E) () delay(2);
0x9b	Draw a filled tip leftward Triangle	Send 0x8b Send the X coordinate of the tip Send the Y coordinate of the tip Send the Y coordinate of the tip Send the With of the tip to the right Send to 10 for display mode (I for positive, 0 for negative) SWalt until receive a module available byte (E') from LED T. Wait 2ms	} void Draw, Filled_Triangle_Left_Ward(int X0_Pos, int Y0_Pos, int width, int negative) { Serial wint(E0/8b); Serial wint(E0/Pos); S
0x9c	Draw a 5p rightward Triangle	Send 0x6c Send bx6c Send bx6c xooxidinate of the tip Send bx xooxidinate of the tip to the left Send to ro for display mode (1 for positive, 0 for negative) Send to ro for display mode (1 for positive, 0 for negative) Send to ro for display mode (1 for positive, 0 for negative) Xivative xooxidinate	void Draw_Triangle_Right_Ward(int X0_Pos, int Y0_Pos, int width, int negative) { Serial write(X0,C): Serial write(Y0,Pos): Serial
0x9d	Draw a filled tip rightward Triangle	Send 0x8d Send was Accordinate of the tip Send the Accordinate of the tip Send the Accordinate of the tip Send the width of the tip to the let Send the width of the tip to the let Send to of to display mode (1 for positive, 0 for negative) Send to of the Accordinate available byte (E) from LED 7. Wait Zms.	/ unid Draw_Filled_Triangle_Right_Ward(int X0_Pos, int Y0_Pos, int width, int negative) { Simila winte(Dod); While (Simila mod) (I=E) () debty(2);
0x9e	Set a pixel for positive display (show pixel)	Send 0x9e Send the X coordinate of the pixel Send the Y coordinate of the pixel Send the Y coordinate of the pixel Wait until receive a module available byte (E) from LED S. Wait 2ms	// void Set_Pixel(int X0_Pos, int Y0_Pos) {
0x9f	Set a pixel for negative display (clear pixel)	Send 0x9f Send the X coordinate of the pixel Send the Y coordinate of the pixel Send the Y coordinate of the pixel Wait until receive a module available byte (E) from LED S. Wait 2ms	void Clear, Pixel (int X0, Pos, int Y0, Pos) { Serial winte(D09); Serial winte(X0, Pos); Serial winte(X0, Pos); Serial winte(X0, Pos); winte(X0, Pos); winte(X0, Pos); delay(2); delay(2); }
0xa0	Display image row by row Up Ward	Send 0xa0 Send 1xe0 Send the speed (typical time is 20ms) Wat until receive a module available byte (E) from LED Wat 2ms	void Display,Row_By_Row_Up_Ward(int Speed) { Serial write(Daol); Serial write(Daol); Serial write(Speed); while (Serial road) !='E') {} delay(2);
0xa1	Display image row by row Down Ward	Send 0xs1 Send 0xs1 Send the speed (typical time is 20ms) Swait until receive a module available byte ('E') from LED Wait 2ms	Judio Display, Row, By, Row, Down, Wardi (int speed) { Serial write(Dos 1); Serial write(speed); while (Serial: read(o) I=E) () delay(2);
0xa2	Display image column by column Left Ward	Send 0xa2 Send the speed (typical time is 20ms) SWait until receive a module available byte (E') from LED 4. Wait 2ms) und Display, Column, By _Column_Left, Ward(int speed) (Serial wint(Cloap); Serial
0xa3	Display image column by column Right Ward	1. Send Dxa3 2. Send the speed (typical time is 20ms) 3.Wait until receive a module available byte (E') from LED 4. Wait 2ms	vaid Display, Column, By, Column, Right, Ward(int Speed) { Serial write(Day); Serial write(Speed); while (Serial.read)) != E) () delay(2);
0xa4	Erase image row by row Up Ward	Send Osa4 Send the speed (typical time is 20ms) Swalt until receive a module available byte ('E') from LED Walt 2ms	red Erace, Row, By, Row_Up, Ward(int Speed) { Serial swint(Cosed): Serial swint(Speed): Serial swint(Speed): while (Serial.read) != E) () delay(2);
0xa5	Erase image row by row Down Ward	Send 0xa5 Send 0xa5 Send 0xa5 Send the speed (typical time is 20ms) Swet until receive a module available byte (E) from LED Wait 2ms	void Erase, Row_By_Row_Down_Ward(int Speed) { Serial wint(Das5); Serial wint(Das5); Serial wint(Das5); Serial wint(Das5); Serial read1) l= E) () Idelay(2);
0xa6	Erase image column by column Left Ward	Send 0xa6 Send the speed (typical time is 20ms) Swat until receive a module available byte (E) from LED Walt 2ms	/ uoid Erase_Column_By_Column_Left_Ward(int Speed) {
0xa7	Erase image column by column Right Ward	Send 0xa7 Send the speed (typical time is 20ms) SWait until receive a module available byte ('E') from LED 4. Wait 2ms	Judo Erase, Column, By, Column, Right, Ward(int Speed) { Serial write(Dar); Serial write(Dar); Serial write(Speed); while (Serial, read) !="E) () delay(2);
0xa8	Display image Inside Out	Send 0xa8 Send the speed (typical time is 20ms) Wait until receive a module available byte ('E') from LED Wait 2ms	/ und Display, Inside, Out (int Speed) { Serial write(Deal); Serial write(Deal); Serial write(Deal); Serial write(Speed); white (Serial read) i= E) () deblay(2);
0xa9	Display image Outside In	Send 0xa9 Send the speed (typical time is 20ms) Wait until receive a module available byte (E') from LED Wait 2ms	/ Local Display, Outside, Inf Int Speed) { Solid wind(Os0); White Control of the
0хаа	Erase image Inside Out	Send 0xaa Send the speed (typical time is 20ms) Swart met receive a module available byte (E) from LED Wait 2ms	void France Inside, Out (int Speed) { Serial write(Opea); Serial write(Speed); while (Serial.read) (I=E) () delay(Z); }
			

0xab	Erase image Outside In	Send the speed (typical time is 20ms) Wait until receive a module available byte (E) from LED Wait 2ms	void Firse Outside, In(int Speed) { Sorial write(Outs); Sorial write(Option); Sorial write(Speed); while (Sorial.read() (=/E) () } }
0xd0	Clear display	Send Ond Selvat until receive a module available byte ('E') from LED Wait 2ms	vaid Class Display Momeny(void) { Send.write(0xd0); white (Send.read) !='E') () delay(2); } }
0xd1	Show the data in the display memory	Send 0xd1 Wait until receive a module available byte ("E") from LED Wait 2ms	void Show, Display, Momeny (void) { Senial, write(Oot 1); while (Senial.tread() !='E') () }
0xd2	Scroll the whole display upward	Send Ond2 Send the shift time (typical time is 70ms) Wait until receive a module available byte ('E') from LED Wait 2ms	said Scrul, Whole, Display_Memory_Up(int shift time) { Scrial write(CPL); Scrial write(CPL); Scrial write(CPL); Scrial write(CPL); While (Scrial.read() I=E) () delay(2);
0xd3	Scroll the whole display downward	Send 0xt3 Send the shift time (typical time is 70ms) Wat until receive a module available byte (E) from LED Walt 2ms	void Scroil_Whole_Display_Memory_Down(int shift time) { Serial write(DAS); Serial write(DAS); Serial write(DAS); Serial write(DAS); Whole_Serial read) i=E) () delay(2);
0xd4	Scroll the whole display leftward	Send 0xd4 Send the shift time (typical time is 70ms) Wall time (typical time is 70ms) Wall time receive a module available byte ("E") from LED Walt 2ms	/ void Scroil_Whole_Display_Memory_Left(int shift time) { Senial wine(God4); Senial wine(God4); Senial wine(God4); while (Senial read) i=E) () delay(2);
0xd5	Scroll the whole display rightward	Send 0xd5 Send the shift time (typical time is 70ms) Wall time (typical time is 70ms)	Void Sendi Whole, Display, Memory, Right(int shift time) { Secretal shrife(Shift): Serial shrife(Shift time); white (Serial read() !='E') () delay(2); }
0xd6	Scroll the section display upward	1. Send ox66 2. Send the X coordinate of up left corner 3. Send the Y coordinate of up left corner 4. Send the X coordinate of bottom right corner 5. Send the Y coordinate of bottom right corner 5. Send the Y coordinate of bottom right corner 6. Send the shift line (Spical lime is Zoms) 7. Walt until receive a module available byte (E) from LED 8. Walt 2ms	void Scnot. Section, Display, Memory_Up(int X0_Pos, int Y0_Pos, int X1_Pos, int Y1_Pos, int shift time) { Scrial write(X0_Pos); Scrial write(X0_Pos); Scrial write(X1_Pos);
0xd7	Scroll the section display downward	1. Send ot Ar 2. Send the X-coordinate of up left corner 3. Send the Y-coordinate of up left corner 4. Send the Y-coordinate of both or pit corner 4. Send the X-coordinate of both or pit corner 5. Send the Y-coordinate of both or pit corner 6. Send the Send the (Spical time is 70ms) 7. Walt until receive a module available byte (E) from LED 8. Wait 2ms	adid Scroll, Scrollon, Disclay, Memory_Down(int X0_Pos, int Y0_Pos, int X1_Pos, int Y1_Pos, int shift time) { Scroll shrink(PO, Pos); Scroll
0xd8	Scroll the section display leftward	1. Send ox88 2. Send the X-coordinate of up left corner 3. Send the Y-coordinate of up left corner 4. Send the Y-coordinate of bottom right corner 5. Send the Y-coordinate of bottom right corner 6. Send the Y-coordinate of bottom right corner 6. Send the shift the (Spical lime is 20ms) 7. Walt until receive a module available byte (E) from LED 8. Wait 2ms	void Scnol. Section, Display_Memory_Left (int X0_Pos, int Y0_Pos, int X1_Pos, int Y1_Pos, int shift time) { Scrial write(D0_Pos); Scrial write(P0_Pos); Scrial write(P0_Pos); Scrial write(P1_Pos); Sc
0xd9	Scroll the section display rightward	1. Send obtal 2. Send the X-coordinate of up left corner 3. Send the Y-coordinate of up left corner 4. Send the X-coordinate of bottom right corner 5. Send the Y-coordinate of bottom right corner 6. Send the Y-coordinate of bottom right corner 7. Wal tural receive a module available byte (E) from LED 8. Walt 2ms	soid Scruto, Disclay, Memory_Right (int X0_Pos, int Y0_Pos, int X1_Pos, int Y1_Pos, int shift time) { Serial write(Q0,Pos); Serial write(Q0,Pos); Serial write(Y0,Pos); Serial write(Y1_Pos); Serial write(Y1_Pos); Serial write(Y1_Pos); Serial write(Y1_Pos); Serial write(pilit time); write(Serial road) != F) () delay(2); }
Oxda	Display quarter of display memory (Available for ModeO, 1, and 2 only)	Send Orda Send the quarter No. to display Send the quarter No and display Wait until receive a module available byte (E) from LED Module Wait Zms	suid Display, quadrant(int quadrant) { Serial write(Out); Serial write(Out); Serial write(Out); White(Out) Serial write(Out); White(Serial read) != E) () delay(2); }
0xf0	Turn display Off	Send 0xf0 Wait until receive a module available byte ("E") from LED Wait 2ms	vsict Display, Cfft (vsict) { Serial winte(odd); white (Serial.read() !='E' () } }
0xf1	Turn display On	Send 0xf1 2.Wait until receive a module available byte (E) from LED 3. Wait 2ms	seld Display_On(void) (Serial varie(0/t); while (Serial read() !=E') ()
0xf2	Set the brightness of the LED Module	Send 0xf2 Send the level of brightness (0-11) Wall receive a module available byte ('E') from LED Wait 2ms	vaid Str. (Display, Brightness) (Serial varies(02); Serial varies(02)
0xf3	Inverse image	1. Sand 0xf3 2.Wait until receive a module available byte ('E') from LED 3. Wait 2ms	Section Sect
0xf6	Change Instruction mode (0 for Hex Coammand, 1 for AT Command)	Send 0xf6 Send instruction mode 1 Send instruction mode 1 Walt until receive a module available byte ('E') from LED Walt 2ms	Int Change_Display_Mode(int mode) { Serial write(Diff); Serial write(I); Write (Serial write(I); Write (Serial write(I); Write (Serial write(I); Write(Serial write(I); Write(I)
0xf7	Change Display Mode	Send 0xf7 Send 0splay mode (0-12) Send 0splay mode (0-12) Wait until receive a module available byte (E) from LED Wait 2ms	said Change, Display_Mode(int mode) { Serial write(OZ); Serial write(OZ); Serial write(OZ); Serial write(mode); white (Serial read) !=E) () delay(2); } }