

PERCOBAAN 5 :  
KOMUNIKASI SERIAL  
( $\mu$ Controller dengan PC)

## KOMUNIKASI SERIAL ( $\mu$ Controller dengan PC)

---

### TUJUAN :

1. Praktikan mampu membuat dan memahami pemrograman komunikasi serial pada  $\mu$ Controller, baik sebagai penerima atau pengirim data.
2. Praktikan mampu membuat aplikasi interface sederhana dengan program Borland Delphi sebagai penghubung antara  $\mu$ Controller dengan PC secara komunikasi serial .

### ALAT-ALAT :

1. Modul training kit  $\mu$ Controller, beserta kabel interkoneksi (kabel ribbon dan kabel RS-232).
2. PC berbasis MS Windows yang terdapat Hyper Terminal, Reads51 dan Borland Delphi.

### MATERI:

1. Hyper 1 (PC =>  $\mu$ Controller)
2. Hyper 2 ( $\mu$ Controller => PC)
3. Inter 1 (PC =>  $\mu$  Controller, dengan Borland Delphi)
4. Inter 2 ( $\mu$  Controller, dengan Borland Delphi => PC)

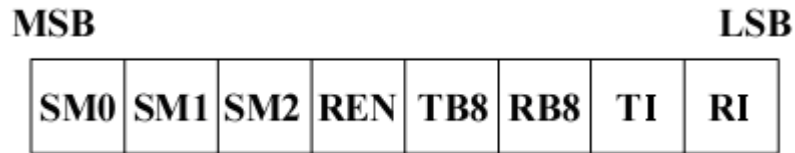
### KONSEP DASAR :

MCS-51 memiliki kemampuan untuk berkomunikasi secara serial melalui pin RXD dan TXD. Satu hal yang perlu diingat adalah tingkat tegangan komunikasi kedua pin serial menggunakan tingkat tegangan TTL. Pada prinsipnya, komunikasi serial adalah komunikasi dimana transmisi data dilakukan per bit. Interface serial hanya membutuhkan jalur yang sedikit (umumnya hanya 2 jalur), sehingga lebih menghemat pin jika dibandingkan dengan interface paralel. Komunikasi serial ada dua macam, asynchronous serial dan synchronous serial :

1. Synchronous serial adalah komunikasi dimana hanya ada satu pihak (pengirim atau penerima) yang menghasilkan clock dan mengirimkan clock tersebut bersama-sama dengan data. Contoh penggunaan synchronous serial terdapat pada transmisi data keyboard.
2. Asynchronous serial adalah komunikasi dimana kedua pihak (pengirim dan penerima) masing-masing menghasilkan clock namun hanya data yang ditransmisikan, tanpa clock. Agar data yang dikirim sama dengan data yang diterima, maka kedua frekuensi clock harus sama dan harus terdapat sinkronisasi. Setelah adanya sinkronisasi, pengirim akan mengirimkan datanya sesuai dengan frekuensi clock pengirim dan penerima akan membaca data sesuai dengan frekuensi clock penerima. Contoh penggunaan asynchronous serial adalah pada Universal Asynchronous Receiver Transmitter (UART) yang

digunakan pada serial port (COM) komputer. MCS-51 mendukung komunikasi secara asinkron, bahkan tiga dari empat serial mode yang dimiliki MCS-51 kompatibel dengan UART.

Register yang digunakan untuk mengatur komunikasi serial terdapat pada Serial Control (SCON)



**Gambar Alokasi Bit SCON**

**Tabel KS-1. Serial Port Control**

<i>Bit</i>	<i>Alamat Bit</i>	<i>Simbol</i>	<i>Deskripsi</i>
SCON.7	9FH	SM0	Pemilih Mode Komunikasi Serial
SCON.6	9EH	SM1	Pemilih Mode Komunikasi Serial
SCON.5	9DH	SM2	Pemilih Mode Komunikasi Multiprosesor
SCON.4	9CH	REN	<i>Reception Enable</i>
SCON.3	9BH	TB8	<i>Bit ke-9 yang Dikirim</i>
SCON.2	9AH	RB8	<i>Bit ke-9 yang Diterima</i>
SCON.1	99H	TI	<i>Transmit Interrupt Flag</i>
SCON.0	98H	RI	<i>Receive Interrupt Flag</i>

Berikut ini adalah penjelasan masing-masing bit SCON yang berkaitan dengan serial port:

- SM0 & SM1  
Pemilih mode komunikasi serial.

**Tabel KS-2. Mode komunikasi serial**

<b>SM0</b>	<b>SM1</b>	<b>Mode</b>	<b>Deskripsi</b>	<b>Baud Rate</b>
0	0	0	8-bit Shift Register	Frek. Osilator/12
0	1	1	8-bit UART	Variabel
1	0	2	9-bit UART	Frek. Osilator/64
1	1	3	9-bit UART	Variabel

Baud rate pada mode 1, 2, dan 3 dapat dilipatgandakan dengan memberi nilai '1' pada SMOD (dalam SFR PCON). Baud rate variabel adalah baud rate yang dihasilkan oleh Timer 1.

- SM2  
Jika SM2 bernilai '1' maka komunikasi multiprosesor diaktifkan dengan kondisi terdapat pada tabel KS-3

**Tabel KS-3. Mode komunikasi**

Mode	Jika SM2 = 1
2 atau 3	R1 tidak akan diaktifkan jika bit ke-9 yang diterima bernilai '0'
1	R1 tidak akan diaktifkan jika stop bit yang valid (bernilai '1') tidak diterima

Pada mode 0, nilai SM2 harus 0

- REN  
REN harus diberi nilai '1' untuk mengaktifkan penerimaan data. Jika REN diberi nilai '0', maka tidak akan ada penerimaan data.
- TB8  
TB8 adalah bit ke-9 yang dikirimkan dalam mode 2 atau 3. Nilai bit ini diatur oleh program user.
- RB8  
RB8 adalah bit ke-9 yang diterima dalam mode 2 atau 3. Pada mode 1, RB8 adalah stop bit. yang diterima. Pada mode 0, RB8 tidak digunakan.

MCS-51 memiliki 4 mode komunikasi serial. Mode 0 berupa synchronous serial (shift register), sedangkan tiga mode yang lain berupa asynchronous serial (UART). Pada semua mode, pengiriman dilakukan jika ada instruksi yang mengisi nilai register SBUF. Sedangkan pada saat penerimaan, data yang diterima akan disimpan pada register SBUF.

#### MODE 0

Mode 0 adalah 8 bit shift register dimana data dikirimkan dan diterima melalui pin RXD sedangkan clock dikirimkan dan diterima melalui pin TXD. Pengiriman data 8 bit dilakukan dengan mengirimkan Least Significant Bit (LSB) terlebih dahulu. Pada mode 0, baud rate yang digunakan adalah sebesar 1/12 dari frekuensi osilator.

#### MODE 1

Pada mode 1, jumlah data yang dikirimkan sebanyak 10 bit yang terdiri dari start bit, 8 bit data (LSB terlebih dahulu), dan stop bit. Ada proses penerimaan, nilai stop bit akan dimasukkan ke RB8 secara otomatis. Pada proses pengiriman, stop bit akan diberi nilai '1' secara otomatis. Pada mode 1, baud rate yang digunakan dapat diatur melalui Timer 1.

#### MODE 2

Pada mode 2, jumlah data yang dikirimkan sebanyak 11 bit yang terdiri dari start bit, 8 bit data (LSB terlebih dahulu), bit ke-9, dan stop bit. Pada proses pengiriman, nilai bit ke 9 dapat diatur dengan mengisi nilai

TB8. Pada proses penerimaan, bit ke 9 akan dimasukkan ke RB8 secara otomatis. Pada mode 2, baud rate yang dapat digunakan adalah sebesar  $1/64$  frekuensi osilator atau  $1/32$  frekuensi osilator jika SMOD bernilai '1'.

### MODE 3

Mode 3 hampir sama dengan mode 2. Perbedaannya terdapat pada baud rate yang digunakan. Jika mode 2 menggunakan baud rate yang pasti, mode 3 menggunakan baud rate yang dihasilkan oleh Timer 1.

Baud rate adalah frekuensi clock yang digunakan dalam pengiriman dan penerimaan data. Satuan baud rate pada umumnya adalah bps (bit per second), yaitu jumlah bit yang dapat ditransmisikan per detik. Baud rate untuk mode 0 bernilai tetap dengan rumus yang terdapat pada persamaan 1.

$$\text{Baud rate} = \frac{\text{Frekuensi Osilator}}{12} \quad (1)$$

Sedangkan baud rate untuk mode 2 memiliki 2 variasi tergantung dari kondisi SMOD. Rumus baud rate untuk mode 2 terdapat pada persamaan 2.

$$\text{Baud rate} = \frac{2^{\text{SMOD}}}{64} \times \text{Frekuensi Osilator} \quad (2)$$

Baud rate untuk mode 1 dan 3 dihasilkan oleh Timer 1. Pengaturan baud rate untuk mode 1 dan 3 dapat dilakukan dengan cara mengubah nilai SMOD, TMOD, dan TH1. Nilai baud rate dapat diperoleh dengan menggunakan persamaan 3.

$$\text{Baud rate} = \frac{2^{\text{SMOD}}}{32} \times \text{Timer 1 Overflow Rate} \quad (3)$$

Umumnya Timer 1 dioperasikan pada mode 2 (8-bit Auto Reload) sehingga didapat persamaan 4.

$$\text{Baud rate} = \frac{2^{\text{SMOD}}}{32} \times \frac{\text{Frekuensi Osilator}}{12 \times (256 - \text{TH1})} \quad (4)$$

Berdasarkan persamaan 4 user dapat menghitung berapa nilai TH1 yang dibutuhkan jika diketahui baud rate yang diinginkan dengan persamaan 5.

$$\text{TH1} = 256 - \frac{2^{\text{SMOD}} \times \text{Frekuensi Osilator}}{384 \times \text{Baud rate}} \quad (5)$$

Satu hal yang harus diperhatikan dalam pengaturan baud rate adalah nilai baud rate dan nilai TH1 diusahakan harus tepat dan bukan merupakan pembulatan. Untuk komunikasi serial kecepatan tinggi, pembulatan terhadap nilai-nilai tersebut dapat mengakibatkan kekacauan dalam proses pengiriman atau penerimaan. Jika terdapat nilai pecahan, user disarankan untuk mengganti osilator dengan frekuensi yang sesuai. Untuk komunikasi dengan kecepatan rendah, toleransi terhadap kesalahan cukup besar sehingga pembulatan masih boleh dilakukan.

Misalkan baud rate yang diinginkan adalah 19200 bps dengan frekuensi osilator 11,0592 MHz. Dengan memasukkan data ini ke dalam persamaan 5 maka akan didapat persamaan 6.

$$TH1 = 256 - (2^{SMOD} \times 1,5) \quad (6)$$

Jika  $2^{SMOD}$  bernilai '1', maka akan didapat TH1 sebesar 254,5. Untuk menghindari TH1 berupa pecahan,  $2^{SMOD}$  harus bernilai '2' (SMOD bernilai '1') sehingga didapat TH1 sebesar 253 atau FDh.

Untuk mendapatkan baud rate yang lambat, user dapat mengoperasikan Timer 1 pada mode 1 dengan rumus pada persamaan 7.

$$\text{Baud rate} = \frac{2^{SMOD}}{32} \times \frac{\text{Frekuensi Osilator}}{12 \times (65536 - \text{Timer 1})} \quad (7)$$

**Tabel KS-4. Nilai dan konfigurasi Baud Rate**

Serial		Timer 1				
Mode	Baud Rate	Frekuensi Osilator	SMOD	C/T	Mode	Reload
0	1,6667 Mbps (max.)	20 MHz	X	X	X	X
2	625 Kbps (max.)	20 MHz	1	X	X	X
1, 3	104,1667 Kbps (max.)	20 MHz	1	0	2	FFh
1, 3	19,2 Kbps	11,0592 MHz	1	0	2	FDh
1, 3	9,6 Kbps	11,0592 MHz	0	0	2	FDh
1, 3	4,8 Kbps	11,0592 MHz	0	0	2	FAh
1, 3	2,4 Kbps	11,0592 MHz	0	0	2	F4h
1, 3	1,2 Kbps	11,0592 MHz	0	0	2	E8h
1, 3	137,5 bps	11,9856 MHz	0	0	2	1Dh
1, 3	110 bps	6 MHz	0	0	2	72h
1, 3	110 bps	12 MHz	0	0	1	FEEDh

Proses inisialisasi bertujuan untuk menentukan mode komunikasi serial dan baud rate yang digunakan. Register yang harus diatur terlebih dahulu meliputi:

### 1. SCON

Langkah pertama adalah menentukan mode yang akan digunakan (mode 0, 1, 2, atau 3), kemampuan menerima data, dan nilai bit ke-9. Misalnya mode yang digunakan adalah mode 1 dengan kemampuan menerima data namun tanpa komunikasi multiprosesor, maka instruksinya adalah sebagai berikut:

```
MOV  SCON, #01010000b
atau
MOV  SCON, #50h
Atau
SETB SM1
SETB REN
```

### 2. TMOD, TH1 dan/atau TL1, PCON, dan TCON

Jika komunikasi serial digunakan dalam mode 1 atau 3, maka langkah berikutnya adalah menentukan baud rate. Misalnya Timer/Counter 1 digunakan sebagai timer dalam mode 2 untuk membangkitkan baud rate 19200 bps, maka instruksinya adalah sebagai berikut :

```
MOV  TMOD, #20h
MOV  TH1, #0FDh
MOV  PCON, #80h
MOV  TCON, #40h
Atau

MOV  TMOD, #20h
MOV  TH1, #0FDh
MOV  PCON, #80h
SETB TR1
```

### 3. IE dan/atau IP

Jika komunikasi serial yang diprogram akan digunakan sebagai sumber interrupt, maka IE dan/atau IP juga harus diatur. Misalnya komunikasi serial digunakan sebagai sumber interrupt dengan prioritas tinggi, maka instruksinya adalah sebagai berikut :

```

MOV    IP, #00010000b
MOV    IE, #10010000b
atau
MOV    IP, #10h
MOV    IE, #90h
atau
SETB   PS
SETB   ES
SETB   EA

```

## Serial Communication Fixed Data

### ***Kirim Data, dari uCon ke ... :***

```

Mov SBUF, A
SCON          = 40h
Tmod          = 20h
th1           = Baudrate
PCON          = x
Clr ti

```

### ***Terima data, dari ... ke uCon :***

```

Mov A, SBUF
SCON          = 50h
Tmod          = 20h
th1           = Baudrate
PCON          = x
Clr ri

```

### ***Baudrate :***

Lihat table KS-4



## PROSEDURE PERCOBAAN & DATA PENGAMATAN :

### 1. Hyper 1 (PC => Mikrokontroler)

```
#include <sfr51.inc>
awal:
cseg
org 00h
sjmp start
;
;interrupt penerimaan
org 23h
clr ri
mov a,sbuf
mov p2,a
reti
;interrupt diaktifkan, serial berada pd mode 1 dg ren = 1
;inisialisasi baudrate
start:
mov sp,#30h
mov scon,#50h
mov tmod,#20h
mov th1,#0fdh          ;baudrate 9600 bps
mov tl1,#0fdh
mov pcon,#00h
setb tr1
setb es
setb ea
sjmp awal
;
end
```

- Buka Program Reads51
- Ketik syntax diatas pada program Reads51, save pada D:\PASKOM\Serial\Hyper1.asm, kemudian compile.
- Bila tidak terdapat pesan kesalahan buka program IspPgm.exe, dan lakukan flashing terhadap file tersebut ke dalam uController.
- Buka program Hyper Terminal (Name = hyper1, Connection Using = Com 1, Bits per second = 9600, Flow control = none).
- Tekan tombol keyboard dan amati nyala led. Catat hasil outputnya
- Setelah selesai melakukan percobaan, nonaktifkan Hyper Terminal dengan meng-KLIK icon DISCONNECT (📞). Jangan meng-EXIT Hyper Terminal.

Output Hyper 1 :

Jika ditekan huruf 'a' :

--	--	--	--	--	--	--	--

Jika ditekan angka 'b' :

--	--	--	--	--	--	--	--

Jika ditekan angka '1' :

--	--	--	--	--	--	--	--

--

 = mati

--

 = menyala

(led menyala = 0, led mati = 1)

Kesimpulan :

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Latihan Mandiri (Buat program untuk membuat tampilan nyala led, bila ditekan 'a' pada keyboard maka led akan menyala dari kiri kekanan, dan bila ditekan 'b' pada keyboard maka led akan menyala dari kanan ke kiri). Save di folder yang sama dengan nama : LatMan Hyper1.asm

```
#include<sfr51.inc>
awal :
cseg                      // Code Segment //
;
org 00h                   // alamat 00h //
sjmp start                // lompat ke Inisialisasi //
;
; interrupt penerimaan
;
org 23h                   // alamat 23h //
awal1 :                   // Kirim data ke Mikon //
clr ri
mov a,sbuf
```

```

sjmp satu
;
satu :
cjne a,# ....h,dua      // Data ASCII untuk "a" //
;
knn :
mov a,#7fh              // Led kiri menyala //
mov ....,a              // Menampilkan di Port 2 //
acall ....              // Delay //
;
knn1 :
.... a                  // Geser ke kanan //
mov p2,a
acall ....              // Delay //
cjne a,#0feh,knn1
sjmp awal1
;
dua :
cjne a,# ....h,awal1    // Data ASCII untuk "b"//
;
kiri :
mov a,#0feh             // Led kanan menyala //
mov ....,a              // Menampilkan di Port 2 //
acall ....              // Delay //
;
kiri1 :
.... a                  // Geser ke kiri //
mov p2,a
acall ....              // Delay //
cjne a,#7fh,kiri1
sjmp awal1
;
; Inisialisasi serial
;
start :
mov sp,#30h
mov scon,#50h
mov tmod,#20h
mov th1,#0fdh
mov tl1,#0fdh
mov pcon,#00h
setb tr1
setb es
setb ea
sjmp awal
;
delay :
mov r0,#07h
ulang:
djnz r2,ulang
djnz r1,ulang
djnz r0,ulang
ret
;
end

```

## 2. Hyper 2 (Mikrokontroler => PC)

```

#include <sfr51.inc>
org 00h
mov p0,#0ffh
mov p1,#0ffh
mov p2,#0ffh
mov p3,#0ffh
;
    mov scon, #01010000b
    mov tmod, #00100000b
    mov th1, #0fdh
    setb tr1
;
acall delay
;
pil : jnb p3.0,start
      sjmp pil
start :
mov a,#41h
lanjut:
acall kirim
mov p2,a
acall delay
inc a
cjne a,#4bh,lanjut
sjmp pil
;
kirim : mov sbuf,a
        jnb ti,$
        clr ti
        ret
;
delay :
    mov r1,#0ffh
    mov r2,#200
    mov r3,#5
ulang :
    djnz r1,ulang
    djnz r2,ulang
    djnz r3,ulang
    ret
end

```

- Buka Program Reads51
- Ketik syntax diatas pada program Reads51, save pada D:\PASKOM\Serial\Hyper2.asm, kemudian compile.
- Bila tidak terdapat pesan kesalahan buka program IspPgm.exe, dan lakukan flashing terhadap file tersebut ke dalam uController.
- Buka program Hyper Terminal (Name = hyper2, Connection Using = Com 1, Bits per second = 9600, Flow control = none).
- Tekan p3.0 pada modul Latih Xtra.

Output Hyper 2 :

.....  
 .....  
 .....  
 .....

Kesimpulan :

.....  
 .....  
 .....  
 .....

Latihan Mandiri (Buat program untuk menampilkan bilangan genap pada Hyper Terminal). Save di folder yang sama dengan nama : LatMan Hyper2.asm :

```
#include<sfr51.inc>
org 00h
mov p0,#0ffh
mov p1,#0ffh
mov p2,#0ffh
mov p3,#0ffh
;
; Inisialisasi Serial
;
mov scon,# 01010000b
mov tmod,# 00100000b
mov th1,# 0fdh
setb tr1
;
acall delay
;
pil :
jnb ....,start          // Jika saklar p3.0 ditekan maka //
sjmp pil
;
start :                  // Menampilkan angka 0 2 4 6 8 //
mov a,#....h            // ASCII dari angka 0 //
lanjut :
acall kirim              // Kirim data ke PC //
```

```

mov p2,a                // Tampil di led //
acall delay
inc a                    // Ditambah 2x //
inc a
cjne a,#...h,lanjut    // Jika ASCII = 3Ah maka //
sjmp pil
;
kirim :                  // Prosedur kirim data dari mikon ke PC //
mov sbuf,a
jnb ti,$
clr ti
ret
;
delay :
mov r1,#0ffh
mov r2,#200
mov r3,#5
;
ulang :
djnz r1,ulang
djnz r2,ulang
djnz r3,ulang
ret
;
end

```

### 3. Inter 1 (PC => Mikrokontroler, dengan Borland Delphi)

Borland Delphi :  
Buat Form Seperti Gambar Di Bawah ;



Pengaturan nilai properties :

Komponen	Properties	Pengaturan
Form1	Caption	Hyper 1
Button1	Caption	Konversi
	Font	12 pt
Button2	Caption	Clear
	Font	12 pt
Button3	Caption	Exit
	Font	12 pt
Comport1	BaudRate	br9600
	Connected	True
Timer1	Enabled	True
	Interval	100
Edit1	Text	

Program :

Double klik pada Button1 :

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  timer1.enabled := true;
end;
```

Double klik pada Button2 :

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  edit1.Text := '';
end;
```

Double klik pada Button3 :

```
procedure TForm1.Button3Click(Sender: TObject);
begin
  close;
end;
```

Double klik pada Timer1 :

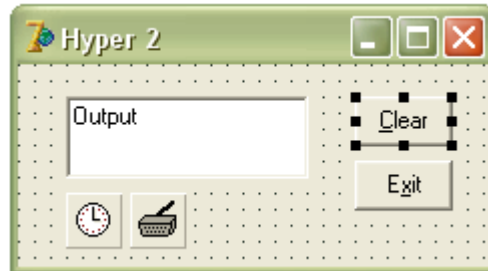
```
procedure TForm1.Timer1Timer(Sender: TObject);
begin
  data:= edit1.Text;
  comport1.writestr (data);
  timer1.enabled := false;
end;
```





#### 4. Inter 2 (Mikrokontroler, dengan Borland Delphi => PC)

Borland Delphi :  
Buat Form Seperti Gambar Di Bawah ;



Pengaturan nilai properties :

Komponen	Properties	Pengaturan
Form1	Caption	Hyper 1
Button1	Caption	Exit
	Font	12 pt
Button2	Caption	Clear
	Font	12 pt
Comport1	BaudRate	br9600
	Connected	True
Timer1	Enabled	True
	Interval	100
Memor1	Name	Output

Program :

Double klik pada Button1 :

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  close;
end;
```

Double klik pada Button2 :

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  Output.Text := '';
end;
```

Double klik pada Timer1 :

```
procedure TForm1.Timer1Timer(Sender: TObject);
var
  str: string;
begin
  comport1.ReadStr(str,1);
  Output.Text := Output.Text + str;
end;
```

- e. Klik File–Save All pada D:\PASKOM\Serial\Project2, Kemudian Tekan tombol F9, cek syntax dan setingan properties bila ada pesan kesalahan.
- f. Lakukan flashing program D:\PASKOM\Serial\Hyper2.hex ke mikrokontroler.
- g. Buka Windows Explore, double klik file exe hasil kompilasi dari program Delphi yang telah dibuat
- h. Jalankan program, perhatikan hasil output led dan buat kesimpulan dari pengamatan kalian

Cat : Bila ada yang kurang jelas, sebaiknya tanyakan kepada asisten yang bertugas.

Output Hyper 2 :

.....

.....

.....

.....

.....

.....

.....

.....

.....

Kesimpulan :

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Paraf Asisten dengan Nama Jelas :

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>M1</b>	<b>M2</b>
----------	----------	----------	----------	-----------	-----------

***Syarat syahnya laporan ini !!!***

**SOAL LAPORAN PENDAHULUAN :**

[illegible]

**JAWABAN LAPORAN PENDAHULUAN :**

[illegible]





### SOAL LAPORAN AKHIR :

[illegible]

**JAWABAN LAPORAN AKHIR :**

This image shows a full page of primary-ruled paper. It features a solid vertical line on the left side, creating a margin. The rest of the page is filled with horizontal dotted lines, providing a guide for handwriting practice. There are no other markings or text on the page.





