วิชา Data Communication Laboratory

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

การทดลองที่ 9 Digital Modulation Techniques

วัตถุประสงค์

1. ศึกษารูปแบบของการแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอก
2. เพื่อเข้าใจวิธีการของมอดูเลชันแบบ ASK และ FSK
3. ศึกษาการมอดูเลตจากการสร้างวงจรภาคส่งและภาครับ
4. ศึกษาผลกระทบของสัญญาณรบกวนต่อผลการแปลงสัญญาณกลับสำหรับแต่ละมอดูเลชันเทคนิค

ทฤษฎี

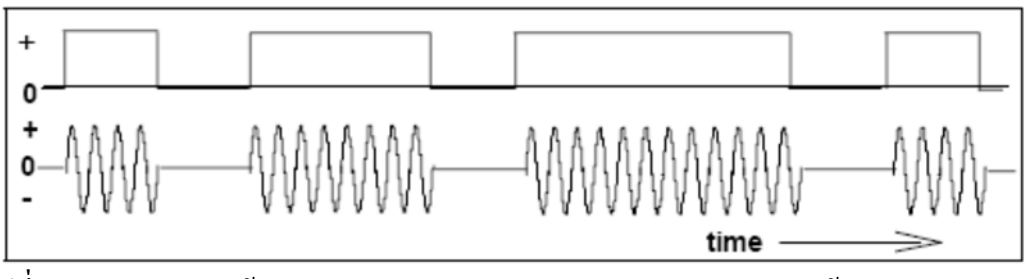
จุดประสงค์ในการมอดูเลตสัญญาณดิจิตอลเพื่อทำให้สามารถส่งข้อมูลดิจิตอลไปได้ระยะทางไกลโดยผ่านช่องสัญญาณที่เหมาะสำหรับการส่งสัญญาณแอนะลอก เทคนิคการมอดูเลตสัญญาณดิจิตอลสามารถทำได้หลายเทคนิค หลักการมอดูเลตสัญญาณดิจิตอล คือ การใช้สัญญาณพาห์ (Carrier Signal) ซึ่งเป็นสัญญาณแอนะลอกนำพาเอาบิตข้อมูลไปยังช่องทางการส่งสัญญาณ เทคนิคการนำพาบิตข้อมูลสามารถทำได้ด้วยการเปลี่ยนคุณสมบัติของสัญญาณพาห์ ได้แก่ ขนาด (Amplitude) ความถี่ (Frequency) และ เฟส (Phase) ตามค่าข้อมูลบิตที่ต้องการแปลงสัญญาณ สัญญาณพาห์ที่นิยมใช้จะเป็นคลื่นไซน์ (Sine Wave) ตัวอย่างเทคนิคการมอดูเลตสัญญาณดิจิตอล ได้แก่ ASK (Amplitude- Shift Keying), FSK (Frequency- Shift Keying), PSK (Phase Shift) และ QAM (Quadrature Amplitude Modulation) เป็นต้น

การมอดูเลตแบบดิจิตอลทางแอมพลิจูด (ASK : Amplitude-Shift Keying)

เป็นการเปลี่ยนค่าขนาดแรงดันของสัญญาณพาห์คลื่นไซน์ ตามบิตข้อมูล

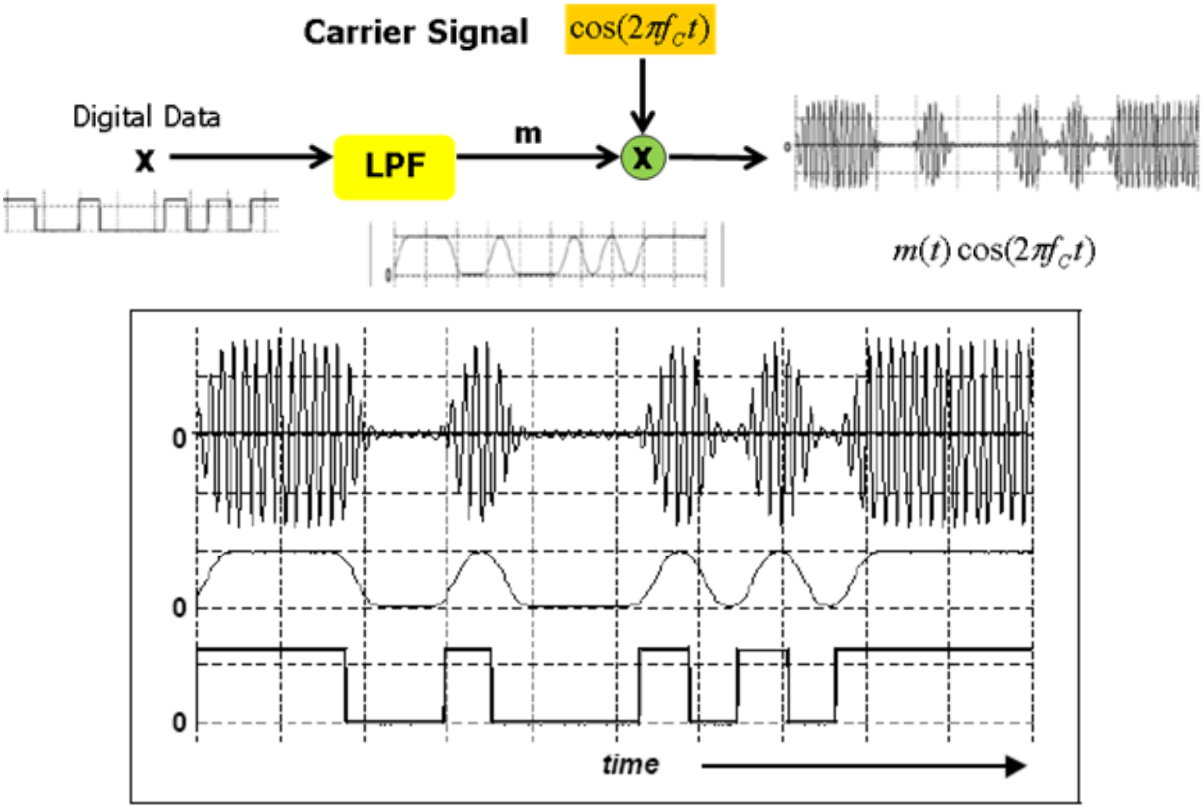
- ค่าบิตข้อมูลเป็น ‘0’ ให้ค่าขนาดแรงดันของสัญญาณพาห์เท่ากับ A_1
- ค่าบิตข้อมูลเป็น ‘1’ ให้ค่าขนาดแรงดันของสัญญาณพาห์เท่ากับ A_2

ตัวอย่างเช่น ให้ $A_1 = 0$ V และ $A_2 = 5$ V ผลการมอดูเลตแบบ ASK เป็นดังรูปที่ 9.1



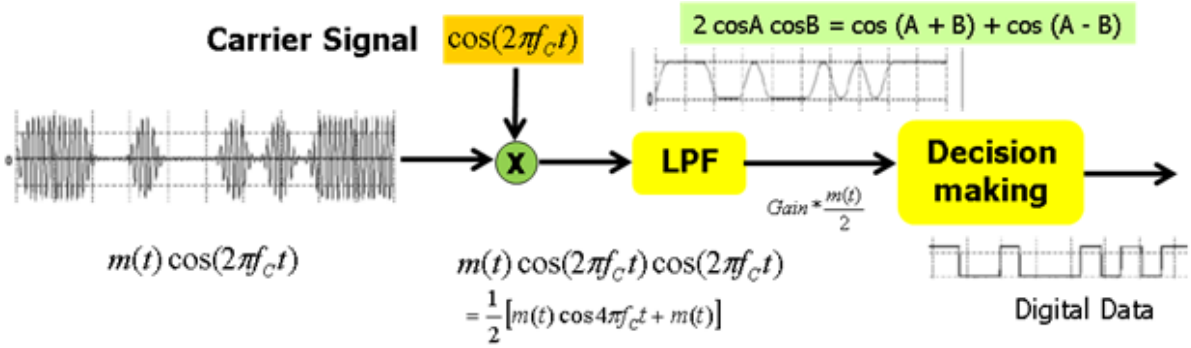
รูปที่ 9.1 แสดงสัญญาณข้อมูลดิจิตอล และสัญญาณการมอดูเลตสัญญาณดิจิตอลด้วยเทคนิค ASK

ในทางปฏิบัติการปรับเปลี่ยนขนาดแรงดันของสัญญาณพาห์ตามค่าบิตข้อมูลทำได้โดยการส่งข้อมูลดิจิตอล ผ่านวงจรกรองความถี่ต่ำ (Low Pass Filter: LPF) หลังจากนั้นนำผลลัพธ์ที่ได้มาปรับขนาด แล้วจึงนำไปคูณกับสัญญาณพาห์ ดังรูปที่ 9.2



รูปที่ 9.2 แสดง Block diagram และ สัญญาณที่ได้จากการมอดูเลตสัญญาณดิจิตอลแบบ ASK ในทางปฏิบัติ

สำหรับขั้นตอนการถอดสัญญาณกลับ หรือ ASK Demodulation ดำเนินการโดยย้อนกลับขั้นตอนของการทำ ASK Modulation ดังรูปที่ 9.3



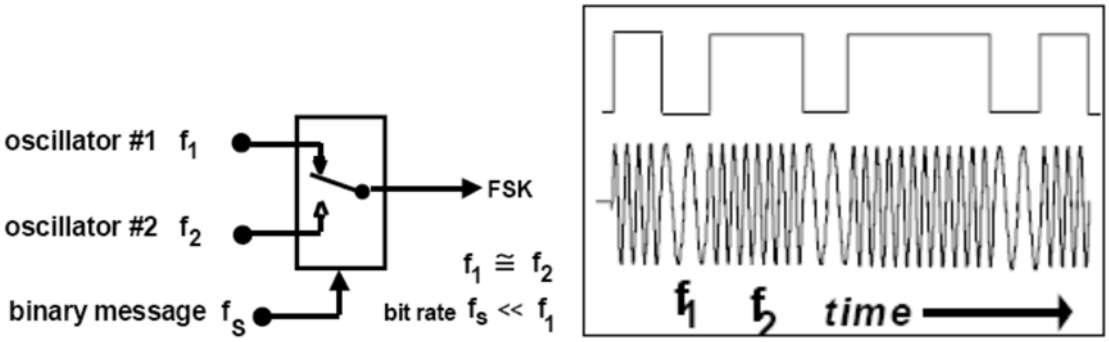
รูปที่ 9.3 แสดง Block diagram สำหรับการทำแปลงสัญญาณกลับสำหรับ ASK Demodulation

การมอดูเลตแบบดิจิตอลทางความถี่ (FSK : Frequency Shift Keying)

เป็นการเปลี่ยนค่าขนาดความถี่ของสัญญาณพาห์คลื่นไซน์ ตามบิตข้อมูล

- ค่าบิตข้อมูลเป็น '0' ให้ความถี่ของสัญญาณพาห์เท่ากับ f_1
- ค่าบิตข้อมูลเป็น '1' ให้ความถี่ของสัญญาณพาห์เท่ากับ f_2

ตัวอย่างผลลัพธ์ของการมอดูเลแบบ FSK ดังแสดงในรูปที่ 9.4



รูปที่ 9.4 แสดงสัญญาณข้อมูลดิจิตอล และสัญญาณการมอดูเลตสัญญาณดิจิตอลด้วยเทคนิค FSK

การมอดูเลตแบบดิจิตอลทางเฟส (PSK : Phase Shift Keying)

การมอดูเลตแบบ PSK เป็นการเปลี่ยนค่าเฟส ของสัญญาณพาห์คลื่นไซน์ ตามบิตข้อมูล เทคนิคที่ง่ายที่สุดสำหรับมอดูเลตแบบ PSK คือ BPSK (Binary Phase Shift Keying) หรือ PRK (Phase Reversal Keying) หรือ Biphase Modulation เป็นการมอดูเลตสัญญาณดิจิตอล โดยที่เปลี่ยนข้อมูลดิจิตอลเป็นสัญญาณแบบสองขั้ว (Bipolar) แล้วทำการมอดูเลตกับสัญญาณคลื่นพาห์ ลักษณะของสัญญาณ BPSK เป็นดังนี้

$$s(t) = Am(t)\cos 2\pi f_c t \;;\; 0 \le t \le T \qquad \text{สมการที่ 9-1}$$

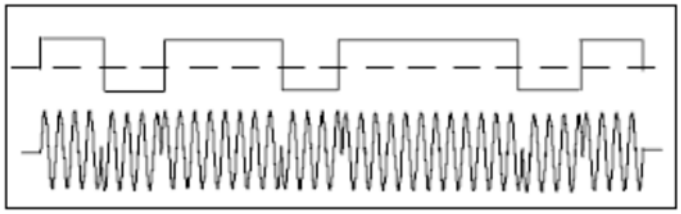
โดยที่ A คือค่าคงที่

$m(t)$ คือสัญญาณอินพุทมีค่า +1 และ –1

f_c คือความถี่ของสัญญาณคลื่นพาห์

T คือช่วงเวลาของบิต

ด้วยการมอดูเลตแบบ BPSK นั้นเอาท์พุทที่จะเป็นไปได้เพียงสองเฟสโดยที่มีสัญญาณคลื่นพาห์เพียงความถี่เดียว โดยเอาท์พุทตัวแรกจะเป็นตัวแทนของสัญญาณไบนารี "1" และเอาท์พุทตัวที่สองจะเป็นตัวแทนของสัญญาณไบนารี "0" ดังรูปที่ 9.5



รูปที่ 9.5 แสดงสัญญาณข้อมูลดิจิตอลแบบสองขั้ว และสัญญาณการมอดูเลตสัญญาณดิจิตอลด้วยเทคนิค BPSK

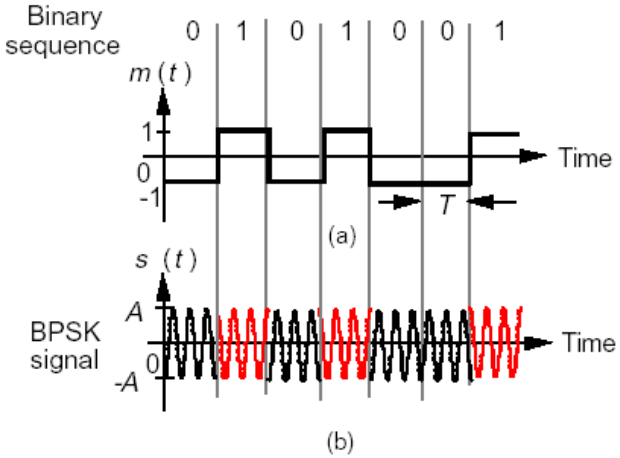
เมื่อสัญญาณอินพุทที่เป็นสัญญาณดิจิตอลมีการเปลี่ยนสถานะ (จาก “0” เป็น “1” หรือ จาก “1” เป็น “0”) ทำให้เอาท์พุทเปลี่ยนเฟสไป 180° ซึ่งทำให้แทนลักษณะการมอดูเลตสัญญาณดิจิตอลด้วยเทคนิค BPSK ได้ดังนี้

- ค่าบิตข้อมูลเป็น ‘0’ ให้มุมเลื่อนของสัญญาณพาห์เท่ากับ π
- ค่าบิตข้อมูลเป็น ‘1’ ให้มุมเลื่อนของสัญญาณพาห์เท่ากับ 0

จากที่กล่าวมาสามารถเขียนสมการของการมอดูเลตสัญญาณดิจิตอลด้วยเทคนิค BPSK อีกรูปแบบได้ดังนี้

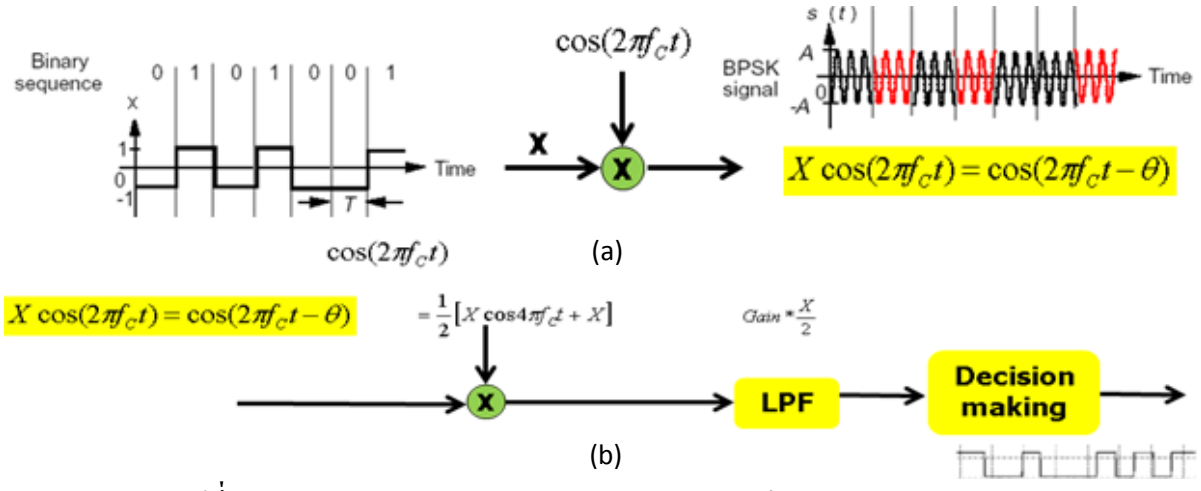
$$s(t) = \begin{cases} A\cos(2\pi f_c t); \ binary \ \ 1 \\ A\cos(2\pi f_c t + \pi); \ binary \ \ 0 \end{cases}$$ สมการที่ 9-2

ในรูปที่ 9.6 (a) นั้นเป็นการแสดงสัญญาณดิจิตอลที่มีสัญญาณข้อมูลเป็น 0 1 0 1 0 0 1 เมื่อทำการมอดูเลตแบบ BPSK จะได้สัญญาณเอาท์พุทในดังรูปที่ 9.6(b)



รูปที่ 9.6 แสดงสัญญาณข้อมูลดิจิตอลแบบสองขั้ว และสัญญาณการมอดูเลตสัญญาณดิจิตอลด้วยเทคนิค BPSK

บล็อกไดอะแกรมการมอดูเลตแบบ BPSK แสดงดังรูปที่ 9.7 (a) บล็อกไดอะแกรมของการดีมอดูเลตแบบ BPSK แสดงดังรูปที่ 9.7 (b)



รูปที่ 9.7 แสดง Block diagram สำหรับการมอดูเลตและดีมอดูเลตแบบ BPSK

การดีมอดูเลตสัญญาณ BPSK

การมอดูเลตแบบ BPSK สามารถทำได้โดยการนำสัญญาณพาห์คูณกับสัญญาณที่รับเข้ามาได้ซึ่งสามารถเขียนเป็นสมการได้ดังนี้

$$r(t) = \left[Am(t)\cos 2\pi f_c t \right]\cos 2\pi f_c t$$
$$= \frac{1}{2} Am(t)\cos 4\pi f_c t + \frac{1}{2} Am(t)$$

สมการที่ 9-3

เมื่อผ่านวงจรฟิลเตอร์แบบความถี่ต่ำผ่านจะได้สัญญาณ $\frac{1}{2} Am(t)$ ซึ่งเป็นสัญญาณที่สามารถแปลงกลับเป็นไบนารีได้

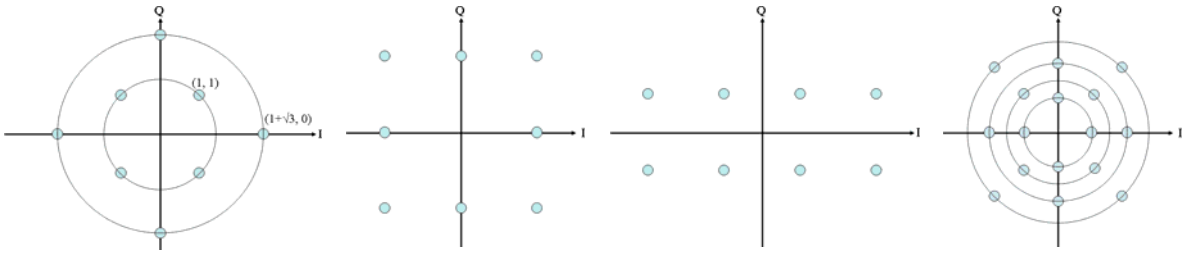
การมอดูเลตแบบ QAM

เป็นการเปลี่ยนคุณสมบัติของสัญญาณพาห์คลื่นไซน์ตามบิตข้อมูล 2 คุณลักษณะคือ ค่าขนาดแรงดัน และมุมเฟส สามารถเลือกเงื่อนไขการเปลี่ยนค่าขนาดแรงดัน และมุมเฟสตามลักษณะของ QAM เช่น 8-QAM สามารถเลือกค่าขนาดและมุมได้หลายแบบ เช่น มี 1 ค่าขนาดแรงดัน และ 8 มุมเฟส หรือ มี 2 ค่าขนาดแรงดัน และ 4 มุมเฟส ได้เช่นกัน

ซึ่งค่าขนาดและมุมเลื่อนที่ถูกเลือกใช้สำหรับ QAM สามารถแสดงได้ในกราฟ Constellation diagram โดยรัศมีของพิกัดของแต่ละจุดจากจุดศูนย์กลางของ Constellation diagram แสดงถึงขนาดของสัญญาณพาห์ และ มุมของพิกัดแต่ละจุดเป็นมุมเลื่อนของสัญญาณพาห์นั่นเอง ตัวอย่างของ Constellation diagram ของ 8-QAM และ 16-QAM แสดงในรูปที่ 9.8

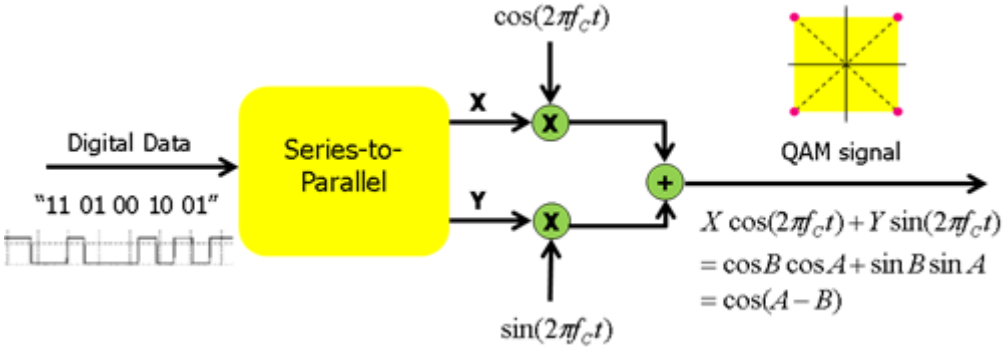
สำหรับเทคนิคการสร้างสัญญาณ QAM ทำได้ดังแสดงในรูปที่ 9.9 โดยทางภาคส่งจะทำการสร้างสัญญาณ QAM หนึ่งชุดที่เป็นตัวแทนข้อมูล 2 บิต จึงมีส่วนของการแปลง serial-to-parallel มาช่วย เพื่อทำให้สามารถส่งสัญญาณ 2 บิต (X และ Y)ไปบนสัญญาณพาห์ที่ความถี่เดียวกัน โดยเอาสัญญาณบิตที่ 1 (X) คูณกับสัญญาณพาห์ที่เป็นสัญญาณ cosine ส่วนบิตที่ 2 (Y) จะถูกคูณกับสัญญาณพาห์ cosine ที่เลื่อนไป 90 องศา นั่นคือ สัญญาณพาห์ sine นั่นเอง จากนั้น สัญญาณของทั้งสองบิตจะถูกรวมเพื่อส่งออกไปพร้อมกันเป็นสัญญาณ QAM

ส่วนที่ภาครับจะนำสัญญาณ QAM ที่ได้รับ มาทำการถอดสัญญาณบิต X และ บิต Y ทีละบิต แล้วจึงจัดเรียงลำดับบิตข้อมูลส่งออกไป (Parallel-to-Serial) โดยการถอดบิตข้อมูลบิต X สามารถทำได้ด้วยการนำสัญญาณ QAM คูณกับสัญญาณพาห์ cosine อย่างไรก็ดี เมื่อสัญญาณเดินทางผ่านช่องนำสัญญาณ สัญญาณอาจมีการเลื่อนตัว ทำให้สัญญาณ QAM ที่ได้รับอาจเลื่อนไปจากที่ภาคส่งส่งออกมา ดังนั้น จึงต้องมีการ Sync สัญญาณพาห์ cosine ของภาครับให้ตรงกับภาคส่ง โดยปรับเลื่อนให้สัญญาณพาห์ตรงกับสัญญาณ QAM ด้วย phase shifter จากนั้น กรองสัญญาณด้วน LPF เพื่อกำจัดสัญญาณรบกวน สุดท้ายจะต้องมีการตัดสินใจว่าสัญญาณที่ได้รับ ควรเป็นข้อมูล ‘0’ หรือ ‘1’ ซึ่งทำได้โดยใช้ Decision Maker สุดท้ายจะได้สัญญาณบิต X กลับออกมาที่ภาครับ ส่วนขั้นตอนในการถอดข้อมูลบิต Y ทำได้เช่นเดียวกัน

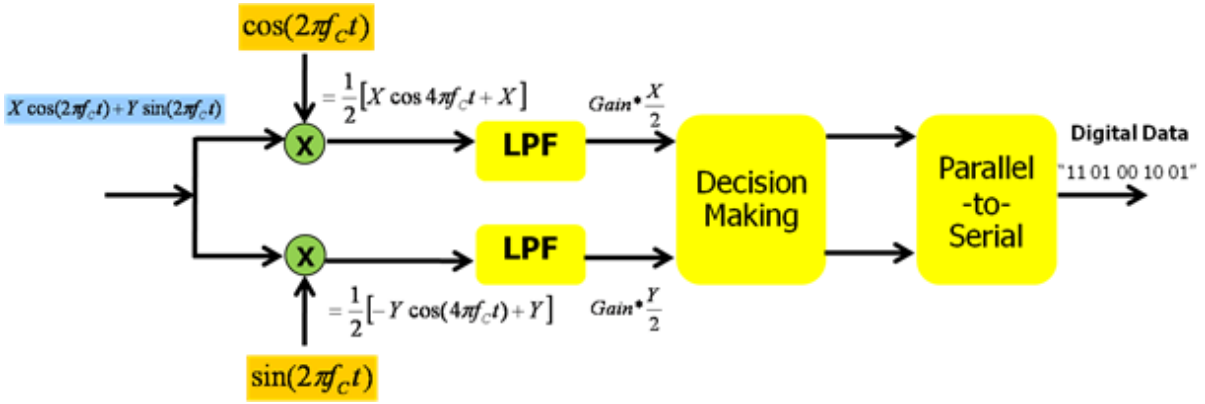


a) Circular 8-QAM b) Rectangular 8-QAM c) Rectangular 8-QAM d) Circular 16-QAM

รูปที่ 9.8 แสดง Constellation Diagram ของการมอดูเลตแบบ QAM



(a) เทคนิคการมอดูเลชั่นแบบ 4-QAM



$$[X\cos(2\pi f_C t)+Y\sin(2\pi f_C t)]\cos(2\pi f_C t)$$
$$=X\cos(2\pi f_C t)\cos(2\pi f_C t)+Y\sin(2\pi f_C t)\cos(2\pi f_C t)$$
$$=X\left(\frac{1}{2}[\cos(0)+\cos(4\pi f_C t)]\right)+Y\left(\frac{1}{2}\sin(4\pi f_C t)\right)$$
$$=\frac{1}{2}[X+X\cos(4\pi f_C t)+Y\sin(4\pi f_C t)]$$

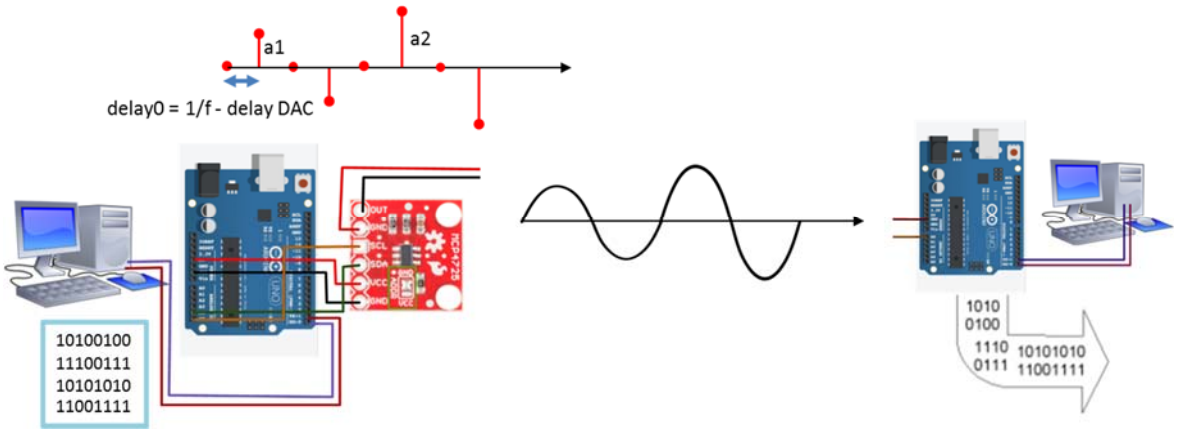
$$[X\cos(2\pi f_C t)+Y\sin(2\pi f_C t)]\sin(2\pi f_C t)$$
$$=X\cos(2\pi f_C t)\sin(2\pi f_C t)+Y\sin(2\pi f_C t)\sin(2\pi f_C t)$$
$$=X\left(\frac{1}{2}\sin(4\pi f_C t)\right)+Y\left(\frac{1}{2}[\cos(0)-\cos(4\pi f_C t)]\right)$$
$$=\frac{1}{2}[X\sin(4\pi f_C t)+Y-Y\cos(4\pi f_C t)]$$

(b) เทคนิคการดีมอดูเลชั่นสัญญาณแบบ 4-QAM

รูปที่ 9.9 Block diagram แสดงเทคนิคการมอดูเลชั่น และดีมอดูเลชั่นสัญญาณแบบ 4-QAM

การทดลองที่ 9.1 ศึกษาการทำงานเบื้องต้นของการแปลงสัญญาณด้วยเทคนิค ASK Modulation และ Demodulation

ให้นักศึกษาต่อวงจรภาคส่ง (Tx) และ ภาครับ (Rx) ตามรูปที่ 9.10 เพื่อทำการรับข้อมูลดิจิตอลชุดละ 8 บิต จากคอมพิวเตอร์ ส่งให้ Arduino ผ่าน Serial Communication แล้วเขียนโปรแกรมที่ Arduino เพื่อทำการแปลงข้อมูลดิจิตอลเป็น Analog Sampling Signal ให้กับ วงจรแปลง Digital-to-Analog Converter (DAC) ซึ่งจะทำการสร้างสัญญาณ Analog Waveform ส่งผ่านสาย ไปยังภาครับ (Rx) ภาครับจะรับสัญญาณเข้าที่ Analog Port ของ Arduino โดยทำการสุ่มวัดค่า (Sampling) และ จัดระดับสัญญาณใหม่ (Quantization) ตามเงื่อนไขของ Analog-to-Digital Converter (ADC) ที่ขา Analog Port ของ Arduino โดย Arduino UNO R3 ที่ใช้ในการทดลอง จะทำการจัดระดับเป็นดิจิตอล 10 บิต ซึ่งมีค่าอยู่ในช่วง [0, 1023] จากนั้นเขียนโปรแกรมที่ Arduino เพื่อนำค่า Amplitude ของ Sampling มาพิจารณาแปลงกลับเป็นข้อมูลดิจิตอล และแสดงผลที่คอมพิวเตอร์ฝั่งรับ



ภาคส่ง (Tx) : PC → Arduino → DAC → ASK Signal **ภาครับ (Rx)** : ASK Signal → Arduino(ADC) → PC

รูปที่ 9.10 รูปแบบการทดลองการแปลงสัญญาณด้วยเทคนิค ASK Modulation และ Demodulation

1. ให้นักศึกษาเขียนโปรแกรมบนบอร์ด Arduino UNO R3 ที่ภาคส่งข้อมูล (Tx) **<u>โดยที่นักศึกษาต้องกำหนดค่าหรือเขียนโปรแกรมในส่วนที่เป็นสีแดงเอง</u>**

   1.1. ส่วนหัวโปรแกรม

   1) กำหนด Library ที่ใช้

```
#include <Wire.h>
#include <Adafruit_MCP4725.h>
Adafruit_MCP4725 dac;
```

   2) กำหนดค่าพารามิเตอร์อื่นๆ

```
#define defaultFreq   1700  //DAC speed (Hz)
/*freq0 : frequency of carrier sine wave (Hz)*/
#define freq0 ____
/*A[0]-A[3] : ASK Amplitude (0,5] (V)*/
const float A[4] = {_____,_____,_____,_____};
/*S_DAC : Amplitude (12bit) of sine wave at 0,90,180,270*/
const uint16_t S_DAC[4] = {_____,_____,_____,_____};
int delay0;
char inData[20];      // Allocate some space for the string
```

1.2. ส่วนฟังก์ชั่น setup()

```
void setup( ) {
  /* set buadrate serial is 115200 */

  dac.begin(0x62);          // set to default
  delay0 = (1000000/freq0 - 1000000/defaultFreq)/4;
    // delay for sampling period of sine
    // (Tsine – delayfrom DAC processing speed)
  Serial.flush();           //  for clear buffer serial
}
```

1.3. ส่วนฟังก์ชั่น loop()ให้เขียนโปรแกรมเพื่อส่งข้อมูลที่ TX

```
void loop( ) {
  if(Serial.available()>0){
    /*use a cycle loop receive inData : message input */
    for (                                              ){
      inData[i] = Serial.read();      // Read a character
    }
    /*use a cycle loop i for send data 8 bits*/
    for (                                              ){
      /*
        use a cycle loop k for 1 ASK signal element (2 bit)
        - map inData[i] to tmp (2 bit)
        - from LSB to MSB
       */
      for (int k=7; k>=0; k-=2){
        int tmp = inData[i] & 3;      // 00, 01, 10, 11
        /*use a cycle loop sl to send 5 cycle/baud*/
        for(                                          ){
          /*use a cycle loop s to send 4 sample/cycle*/
          for(                                         ){
          /*
            Use the selected amplitude above to modify
            sine amplitude
          */
            dac.setVoltage(                    ,false);
            delayMicroseconds(delay0); // sampling delay
          }
        }
        inData[i] >>=2;
      }
    }
    dac.setVoltage(0,false);    // for don't send
  }
}
```

2. ใช้ Oscilloscope ทำการวัดสัญญาณผลลัพธ์ที่ได้หลังจากวงจร DAC
3. เขียนโปรแกรมบนบอร์ด Arduino UNO R3 ที่ภาครับข้อมูล (Rx) เพื่อถอดข้อมูลกลับ **<u>โดยที่นักศึกษาต้องกำหนดค่า หรือเขียนโปรแกรมในส่วนที่เป็นสีแดงเอง</u>**

3.1. ส่วนหัวโปรแกรม

1) กำหนด Library ที่ใช้เช่นเดียวกับข้อ 1.1

2) กำหนดค่าพารามิเตอร์อื่นๆ

```
#define defaultFreq   1700  //DAC speed (Hz)
/*freq0 : frequency of carrier sine wave (Hz)*/
#define freq0 ____
/*A[0]-A[3] : ASK Amplitude (0,5] (V)*/
const float A[4] = {______,______,______,______};
int delay0;
/* amin/amax : Amplitude in digital 10bit */
#define a0min ____     /* a0min <= a0 <= a0max */
#define a0max ____
#define a1min ____     /* a1min <= a1 <= a1max */
#define a1max ____
#define a2min ____     /* a2min <= a2 <= a2max */
#define a2max ____
#define a3min ____     /* a3min <= a3 <= a3max */
#define a3max ____
/* amplitude difference for detecting rising or falling
signal */
#define r_slope ____
int sum = 0;
int max = 0;
int prev = 0;
int check = false;
int output = -1;
int count = 0;
```

3.2. ส่วนฟังก์ชั่น setup()

```
void setup() {
  /* set serial baudrate the same as in TX */


}
```

3.3. ส่วนฟังก์ชั่น loop()ให้เขียนโปรแกรมเพื่อส่งข้อมูลที่ RX (Arduino ADC โดยแปลง Analog Amplitude ให้เป็นดิจตอล 10 bits for each sample ซึ่งจะมีค่าอยู่ในช่วง [0, 1024])

```
void loop() {
  int tmp = analogRead(A0); // read signal from analog pin
  if(tmp-prev> r_slope && check ==false){
    max = 0;
    check = true;    // change check status is true
  }
  if(tmp>max){       // update max value
    max=tmp;
  }
  if(max-tmp > r_slope){  // check for falling signal
    if(check == true){
      if(a0min<max && max<a0max){
        Serial.print("0 0 ");
        count++;
      }
      else if(a1min<max && max<a1max){
        Serial.print("0 1 ");
        count++;
      }
      else if(a2min<max && max<a2max){
```

```
            Serial.print("1 0 ");
            count++;
          }
          else if(a3min<max && max<a3max){
            Serial.print("1 1 ");
            count++;
          }
          if(count == 5){
            Serial.println();
            count = 0;
          }
        }
        check = false;  // change check status is false
      }
      prev = tmp;   // assign temp value to previous
    }
```

4. ทดสอบการรับส่งข้อมูลระหว่างอุปกรณ์
5. คำนวณค่า Bandwidth ของการมอดูเลทแบบ ASK ที่ทำการทดลอง

   _____

6. คำนวณความถี่สูงสุด (freq0) ที่สามารถส่งได้ สำหรับ Arduino ที่ส่งข้อมูล Serial ผ่าน I2C protocol ไปยัง DAC ด้วยความถี่ 1700 Hz

   _____

7. ถ้าต้องการปรับโปรแกรมให้สามารถส่ง 1 Bit ใน 1 Signal Element (Baud) จะต้องแก้โปรแกรมของตัวส่งอย่างไร

   _____

   _____

   _____

8. ให้ อธิบายเหตุผลในการเลือกค่า min, max ในการรับ a0 - a3 และถ้าเลือกค่ามากกว่าหรือน้อยกว่าจะให้ผลการรับข้อมูลเป็นอย่างไร

   _____

   _____

9. ให้วัดเวลาในการทำมอดูเลชันสำหรับการส่ง 2 บิต แต่ละครั้ง (ปรับโปรแกรมให้วัดเวลาโดยคำนวณความต่างของเวลาก่อนและหลังทำมอดูเลชั้น ด้วยคำสั่ง micros()) และอธิบายผลของ การปรับจำนวน bit / cycle ว่าส่งผลอย่างไร ต่ออัตราความเร็วในกการส่งบิตข้อมูล

   _____

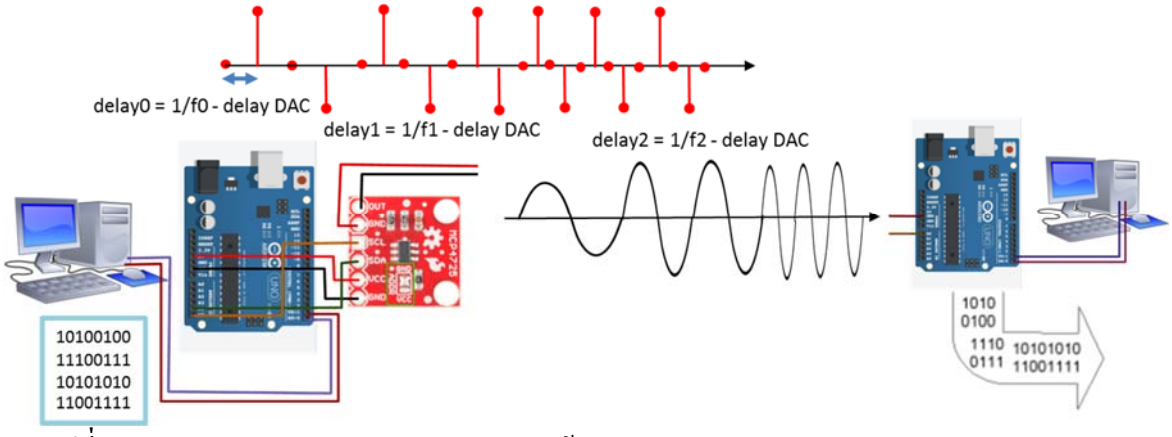
   _____

.....

ลายเซ็นอาจารย์ผู้ตรวจการทดลอง

การทดลองที่ 9.2 ศึกษาการทำงานเบื้องต้นของการแปลงสัญญาณด้วยเทคนิค FSK Modulation และ Demodulation

ในการทดลองนี้จะใช้วงจรที่ต่อไว้ในการทดลองก่อนหน้าดังรูปที่ 9.11 แต่เปลี่ยนการเขียนโปรแกรมในส่วนของการทำ Digital Modulation ให้เป็นแบบ FSK Modulation / Demodulation



รูปที่ 9.11 รูปแบบการทดลองการแปลงสัญญาณด้วยเทคนิค FSK Modulation และ Demodulation

1. ให้นักศึกษาเขียนโปรแกรมบนบอร์ด Arduino UNO R3 ที่ภาคส่งข้อมูล (Tx)

1.1. ส่วนหัวโปรแกรม

1) กำหนด Library ที่ใช้

```
#include <Wire.h>
#include <Adafruit_MCP4725.h>
#include <Adafruit_ADS1015.h>
```

2) กำหนดค่าพารามิเตอร์อื่นๆ

```
#define defaultFreq   1700       // dac speed (Hz)
#define f0 500          // FSK f0
#define f1 750          // FSK f1
#define f2 1000         // FSK f2
#define f3 1250         // ASK f3
int delay0, delay1, delay2, delay3;
/*S_DAC : Amplitude (12bit) of sine wave at 0,90,180,270*/
const uint16_t S_DAC[4] = {______,______,______,______};
Adafruit_MCP4725 dac;
```

1.2. ส่วนฟังก์ชั่น setup()

```
void setup( ) {
  /* set buadrate serial ที่115200 **/
  dac.begin(0x62);    // set to default
  /*
    calculate sampling period (time) of sine[4]
    for each FSK Frequency
  */
  delay0 = _____       //sampling period for FSK 500 Hz
  delay1 = _____       //sampling period for FSK 750 Hz
  delay2 = _____       //sampling period for FSK 1000 Hz
  delay3 =_____        //sampling period for FSK 1250 Hz
  Serial.flush();    //  for clear buffer serial
}
```

1.3. ส่วนฟังก์ชั่น loop()ให้เขียนโปรแกรมเพื่อส่งข้อมูลที่ TX

```
void loop( ) {
  if(Serial.available() > 0){    // for get input
    int in = Serial.parseInt();  // get Dec from Serial
    /*
     create an array store every set of 2 bits
      from each input byte
    */
    /*for each set of 2 bits in a data byte*/
    for (int k=3 ; k>=0 ; k--){
      if(input[k] == 0){         // for input 00 -> 500 Hz
        //display bit value of input [k] (out '00')
        //send S_DAC[4] N cycles (calculating for 500Hz)
      }
      else if(input[k] == 1){    // for input 01 - 750 Hz
        //display bit value of input [k] (out '01')
        //send S_DAC[4] N cycles  (calculating for 500Hz)
      }
      else if(input[k] == 2){    // for input 10 - 1000 Hz
        //display bit value of input [k] (out '10')
        //send S_DAC[4] N cycles (calculating for 1000Hz)
      }
      else if(input[k] == 3){    // for input 11 - 1250 Hz
        //display bit value of input [k] (out '11')
        //send S_DAC[4] N cycles (calculating for 1250Hz)
      }
    }
    dac.setVoltage(0, false);    // for don't send
  }
}
```

2. ใช้ Oscilloscope ทำการวัดสัญญาณผลลัพธ์ที่ได้หลังจากวงจร DAC

3. เขียนโปรแกรมบนบอร์ด Arduino UNO R3 ที่ภาครับข้อมูล (Rx) เพื่อถอดข้อมูลกลับ

3.1. ส่วนหัวโปรแกรม

1) กำหนด Library ที่ใช้เช่นเดียวกับข้อ 1.1

2) กำหนดค่าพารามิเตอร์อื่นๆ

```
/* cbi this for increase analogRead Speed */
#ifndef cbi
#define  cbi(sfr,  bit)  (_SFR_BYTE(sfr)  &=  ~_BV(bit))
#endif
#ifndef sbi
#define sbi(sfr, bit) (_SFR_BYTE(sfr) |= _BV(bit))
#endif
/*amplitude diff. for detecting rising or falling signal*/
#define r_slope ____
```

3.2. ส่วนฟังก์ชั่น setup()

```
void setup() {
  sbi(ADCSRA,ADPS2) ; // this for increase analogRead Speed
  cbi(ADCSRA,ADPS1) ;
  cbi(ADCSRA,ADPS0) ;
  /* set serial baudrate the same as in TX */
}
```

3.3. ส่วนฟังก์ชั่น loop()ให้เขียนโปรแกรมเพื่อส่งข้อมูลที่ RX (Arduino ADC แปลง Analog Amplitude ให้เป็น 10 digital bits for each sample ซึ่งจะมีค่าอยู่ในช่วง [0, 1024])

```
void loop() {
  /* read signal from Analog pin */
  /* check period of input analog signal */
  /* calculate input frequency */
  /* decode data bits from detected input frequency */
  /* show read data bits */
}
```

4. ทดสอบการรับส่งข้อมูลระหว่างอุปกรณ์
5. คำนวณค่า Bandwidth ของการมอดูเลทแบบ FSK ที่ทดลอง

_____

_____

6. ถ้าต้องการปรับโปรแกรมให้สามารถส่งความถี่ 2000 Hz จะทำได้หรือไม่ ถ้าทำได้จะต้องปรับโปรแกรมอย่างไร ถ้าไม่ได้ ให้นศ.อธิบายเหตุผลที่ไม่สามารถทำได้

_____

_____

_____

_____

7. การปรับจำนวน bit / frequency ส่งผลอย่างไร ต่ออัตราการส่งบิต

_____

_____

8. ให้นศ.อธิบายอัลกอริทึมในการตรวจสอบความถี่ของสัญญาณเพื่อนำมาแปลงกลับเป็นบิตข้อมูล

_____

_____

_____

_____

.....

ลายเซ็นอาจารย์ผู้ตรวจการทดลอง