

로봇 축구 시스템의 개발(MIRAGE I) 3

한국현, 최정이, 최필순, 이세중 | KAIST 전기 및 전자공학과 동아리 '미라지'

지능적이고 안전하며 튼튼한 로봇을 만들겠다는 일념으로 밤낮없이 매진한 결과 공식적인 대회에서의 수상을 통해 그 우수성을 검증 받은 축구로봇을 중심으로 하여 본 팀이 구현했던 로봇과 비전 시스템, 그리고 전략 알고리즘 등에 대해 정리를 해 보고자 하며, 이 글이 로봇을 연구하고 시스템 설계를 공부하는 전자 관련 학과 학생들에게 많은 도움이 되길 바란다.

통신

통신은 80C196KC의 TxD, RxD port를 이용한 serial communication으로 Motorola사의 MC2831A (협대역 FM 송신용 IC)와 MC3356(wide band FSK receiver)를 이용하였다. 하나의 PCB board(7.4cm * 7.4cm)에 송신회로와 수신회로를 함께 꾸몄고 46.9MHz와 49.9MHz의 두 주파수를 사용하여 쌍방향 통신이 가능하게 하였다.

하나의 board 내에 있는 송신 회로단과 수신 회로단 사이에서의 신호의 분리는 마쓰시다 전기의 표면파 필터(SAW Filter)를 사용하여 해결하였다. SAW(Surface Acoustic Wave) device는 전기적 에너지를 기계적 에너지로 변환하는 압전 재료의

〈표 4〉 SAW Filter 사양

EFCH46MCN01		EFCH49MCN01	
Frequency (MHz)	Attenuation (dB)	Frequency (MHz)	Attenuation (dB)
Pass Band (46.61MHz-46.97MHz)	6.0dB MAX	Pass Band (49.67MHz-49.99MHz)	6.0dB MAX
Rejection Band (49.67MHz-49.99MHz)	40dB MIN. from Pass Band	Rejection Band (46.41MHz-46.97MHz)	40dB MIN. from Pass Band

표면을 전파하는 파의 성질을 이용한 진동 소자로서 혁신적인 소형화를 이룩할 수 있다는 장점을 가지고 있다.

실제 제작에 사용된 SAW Filter는 우리나라 모 기업 연구소에 계신 분의 도움을 얻어 직접 일본의 마쓰시다 전기에 FAX로 주문하여 sample로 받을 수 있었다. SAW Filter의 간단한 사양은 표 4와 같다. 실제로 제작된 회로는 9600bps의 속도를 얻을 수 있었고 그 동작은 다음과 같다.

• 송신 회로

그림 15의 회로도에서 좌측 부분의 MC-2831A와 주변 부품들이 송신회로를 나타낸다.

MC2831A는 Motorola사에서 개발한 협대역 FM 송신용 IC로서 마이크 앰프, 가변 리액턴스 변조회로, 쿨피츠 발진회로, 톤 스위치가 부가된 파일럿 신호 발진기, 배터리 체크 회로 등을 내장하고 있다. 동작을 살펴

보면, 먼저 마이크로컨트롤러(80C196KC)에서 나오는 RS232C 송신 데이터가 3번 핀의 입력으로 들어가게 된다.

15번과 16번 핀에 연결되어 있는 병렬 공진회로의 소자값을 조절하면 칩 내부 RF 발진기의 발진 주파수를 결정할 수 있고 이 주파수에 의해 입력 신호의 frequency shift가 일어나게 된다. 이렇게 modulation된 신호는 14번 핀을 통해 출력되어 칩 외부의 증폭단을 거치면서 증폭되고 안테나를 통해 전송된다.

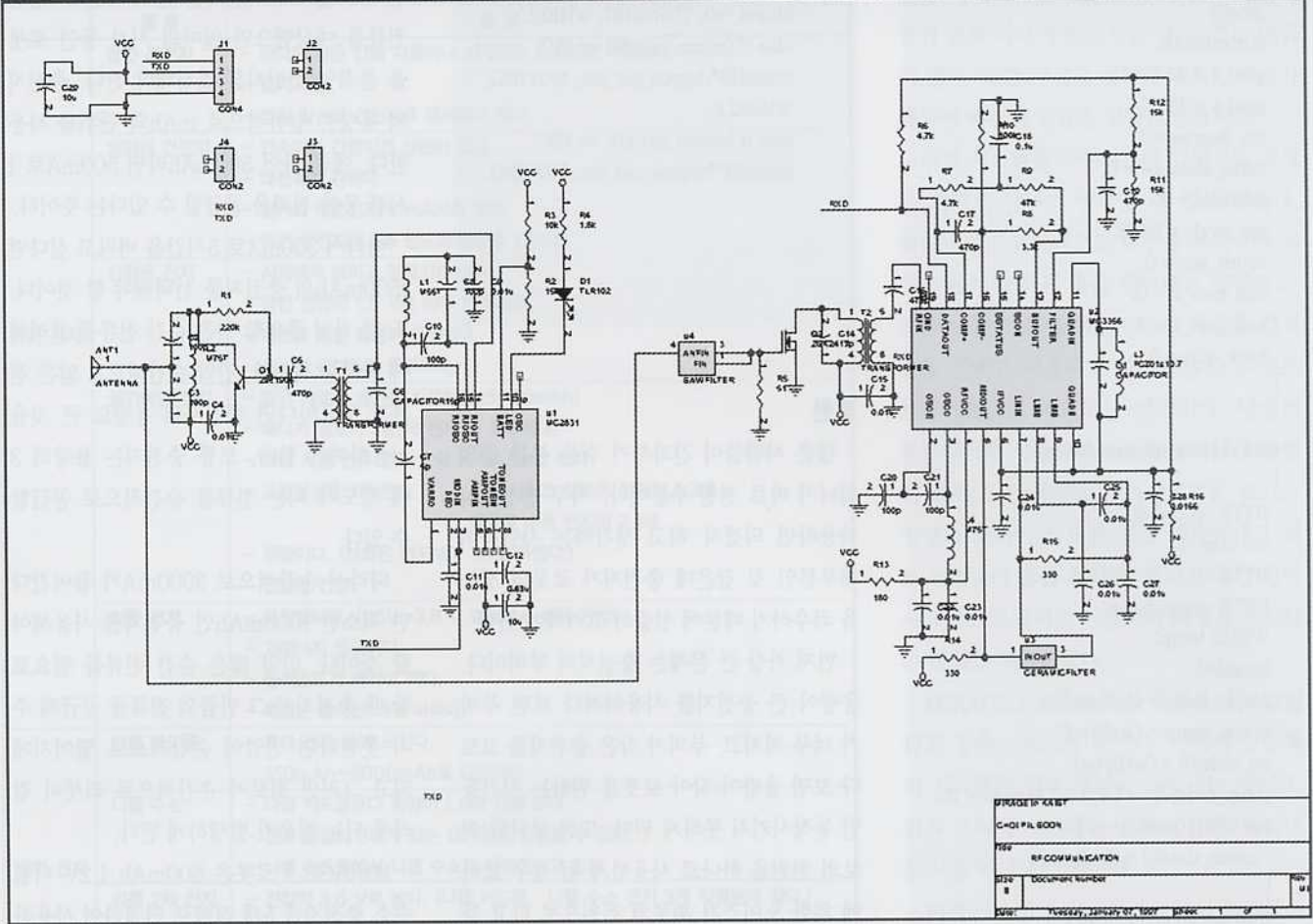
• 수신 회로

그림 15의 회로도에서 우측 부분의 MC-3356과 주변 부품들이 수신회로를 나타낸다. 안테나를 통해 수신된 신호는 표면파 필터를 통해 송신 회로와의 isolation이 이루어지게 되는데, 이는 전력이 큰 송신 신호가 그대로 같은 board 내에 있는 수신회로로 입력되어 회로가 오동작하거나 파손되는 것

〈그림 14〉 로봇의 RF 보드



(그림 15) 통신 보드 회로도



을 방지해 준다.

MC3356은 역시 Motorola사에서 개발한 광대역 FSK 수신기로서 국부발진용 콜피츠 발진회로, 고주파 mixer회로 IF용 Limiting Amp, Quadrature 검파회로, 파형정형용 Comparator 회로 등을 포함하고 있다.

동작을 살펴보면 안테나를 통해 수신된 신호가 필터를 거친 뒤 먼저 증폭단에 의해 증폭이 이루어진다. 20번 핀을 통해 입력된 신호는 고주파 믹서에 의해 IF 신호로 변환되는데 이 외에도 외부 공진회로의 소자값을 조절하여 Local Oscillator의 주파수를 결정할 수 있다. 10.7MHz의 IF 신호로 변환되면 5번 핀을 통해 출력되어 무라다사의 세라믹 필터를 거쳐 신호가 필터링 되고 이 신호가 다시 7번 핀을 통해 입력되어 IF용 Limiting Amp를 거치며 AM 성분이 억제

되고 증폭이 이루어진다.

그 다음은 Quadrature 검파회로를 통해 복조가 이루어지고 Comparator를 거치며 파형정형이 이루어져서 18번 핀을 통해 최종 신호가 나오게 된다. 이 신호는 곧바로 마이크로컨트롤러의 RS232C 수신 데이터로 사용된다.

축구로봇용 소스 코드(일부만 공개)

```
void main(void)
{
    BYTE data;
    init_robot();
    enable();

    while(!start_sig);
    start_sig = 0;
    outbc(0x03);
    int_mask = 0x00;
```

```
position = MY_ID - 1;
while (1) {
    vector_calc();
    state();
    target();
    delta_dis_deg();
    moving();
}

void reset(void)
{
    velocity(GO, STOP);
    wait(3000);
    outbc(0x00);
    asm rst;
}

void init_robot(void)
{
    init_pwm();
    init_sensor();
    init_serial();
    init_PPI(0x8a);
```



```

cosine0;
sine0;
outbc(0x00);
goal_x = 45;
goal_y = 133;
my_degree = 0;
delta_distance = 0;
position = 0;
my_mode = 0xa6;
count_sen = 0;
ball_mov_x = 0;
ball_mov_y = 0;
back_quit = 0;
}

void receive_interrupt(void)
{
    BYTE op_code[1];
    int num=1;
    BYTE i;
    BYTE comm_temp;
    WORD temp;
    disable0;
    comm_temp = GetByte0;
    comm_temp = GetByte0;
    op_code[0] = GetByte0;
    if (op_code[0] == OP_DATA) num = 11;
    for (i=0; i<(num; i++) {
        comm_data[i] = GetByte0;
    }
    if ((receive_error != 1) {
        if (op_code[0] == OP_DATA) {
            set_data0;
        }
        else if (op_code[0] == OP_INST) {
            if (comm_data[0] == START_F) {
                start_sig = 1;
                half = 1;
            }
            else if (comm_data[0] == RESET)
                reset0;
            else if (comm_data[0] == EMER)
                emergence0;
            else if (comm_data[0] == PENAL)
                penalty0;
            else if (comm_data[0] == SHOOT) {
                shoot(MAX_speed, 20);
                velocity(GO, STOP);
                wait(5000);
            }
            else if (comm_data[0] == R30)
                sturn(30, TURNING, WHEEL);
            else if (comm_data[0] == L30)
                sturn(-30, TURNING, WHEEL);
            else if (comm_data[0] == R90)
                sturn(90, TURNING, WHEEL);
        }
    }
}

```

```

else if (comm_data[0] == L90)
    sturn(-90, TURNING, WHEEL);
else if (comm_data[0] == F10)
    sturn(10*degree_per_cm, MOVING, WHEEL);
else if (comm_data[0] == F50)
    sturn(50*degree_per_cm, MOVING, WHEEL);
.
.
.

```

전원

많은 사람들이 간과하기 쉬운 부분 중의 하나가 바로 전원 부분이다. '아무 충전지나 사용하면 되겠다' 하고 생각하는 사람들이 대부분인 것 같은데 충전지가 로봇의 성능을 좌우하기 때문에 신중히 고려해야 한다.

먼저 가장 큰 문제는 충전지의 부피이다. 용량이 큰 충전지를 사용하려다 보면 부피가 너무 커지고, 부피가 작은 충전지를 고르다 보면 용량이 작아 로봇을 원하는 시간동안 동작시키지 못하게 된다. 또한 모터와 회로의 전원을 하나로 사용하게 될 경우 모터에 의한 노이즈가 회로의 전원으로 타고 들어가 회로에서 오동작이 일어날 수 있게 되는 등 예상치 못했던 문제점들이 전원부에서 나타나게 된다.

안정된 전원을 만들기 위해서는 필히 레귤레이터를 달아 회로의 전원으로 사용하여야 한다. 레귤레이터 중에서도 low-drop 레귤레이터를 이용하면 5.5V 정도로도 5V를 낼 수 있다.

충전지를 선택하려면 먼저 로봇이 전류를

얼마나 필요로 하는지를 알아야 한다. 평균 전류를 예상했으면 얼마의 시간 동안 로봇을 움직일 것인지를 결정해야 한다. 충전지의 용량은 일반적으로 mAh의 단위를 사용한다. 예를 들어 500mAh라면 500mA로 1시간 동안 전력을 공급할 수 있다는 뜻이다.

따라서 300mA로 5시간을 버티고 싶다면 1500mAh의 충전지를 선택해야 할 것이다. 또 한 가지 주의할 점은 순간 전류를 고려해야 한다는 점이다. 만약 순간적으로 많은 전류가 필요하다면 충전지의 용량도 큰 것을 사용하여야 한다. 보통 충전지는 용량의 3배 정도가 되는 전류를 순간적으로 공급할 수 있다.

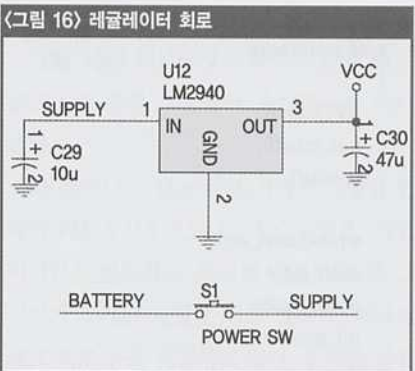
따라서 순간적으로 3000mA가 들어간다면 최소한 1000mAh인 충전지를 사용해야 할 것이다. 만약 많은 순간 전류를 필요로 할 때 충전지가 그 만큼의 전류를 공급해 주지 못한다면 전압이 순간적으로 떨어지게 되고 그러면 회로가 순간적으로 리셋이 걸리게 되는 경우가 발생하게 된다.

MIRAGE I 로봇은 600mAh 1.2V 니켈수소 충전지를 5개 직렬로 연결하여 사용하였다. 따라서 6.0V, 600mAh의 전원을 사용한 것이다. 이것으로 모터와 회로의 전원을 함께 사용하였으며, 회로 전원에는 low-drop 레귤레이터인 LM2940을 사용하였다. MIRAGE I 로봇은 평균 400~500mA의 전류를 소모했으므로 이론상 약 40분 정도 움직일 수 있었다.

참고로 충전지가 아닌 1차 전지도 사용할 수 있을 것이다. 1차 전지를 사용하면 한 번 실험할 때마다 전지를 갈아야 하기 때문에 돈이 많이 든다는 장점이 있지만 충전지가 마땅한 것이 없다면 한 번 고려해볼 만하다. 표 5는 여러 종류의 1차 전지, 2차 전지의 특성을 나타내어 보았다.

PCB Artwork

MIRAGE I 로봇의 모든 보드는 PCB로 제작하였다. 로봇의 크기가 소형화가 되어



〈표 5〉 전지의 특성

	종류	특성
1차 전지	망간 건전지	- 라디오 같은 간헐 사용이나 시계같은 소전류에 적합하다. - 값이 싸다. - 점점 알카리 건전지로 대체되고 있다.
	알카리 건전지	- 연속이건 간헐이건 상관없다. - 대전류에 강하다. - 알카리 버튼전지(10mAh)로 있다. - 망간 건전지의 2배 정도의 용량을 갖는다.
	산화은 전지	- 시계에서 쓰이고 있다.(10mAh) - 방전 전압이 1.5 V로 아주 안정하다. - 대전류 방전 특성이 우수하다. - 초소형 사이즈가 풍부하다.
	공기아연 전지	- 양극 재료로 공기를 사용한다.(5~50mAh) - 에너지 밀도가 기존의 전지 중 최대이다. - 최대 사용 전류 밀도는 약 10mA/cm ² 이다. - 고용량성을 자랑한다. · 동일 사이즈 산화은 전지의 4.3배 · 대용량 수은 전지의 2.1배 - 경량이다. (산화은 전지보다 30% 가볍다.) - 안정한 전압
	리튬 전지	- 고전압이다. (1.5V~3.6 V 범위에서 선택 가능) - 고에너지 밀도이다. - 동작온도가 제약이 적다. - 특성은 좋지만 너무 비싸다.
	니켈 카드뮴	- 휴대 전화 등에 쓰인다. - 150mAh~8000mAh로 다양하다.
2차 전지	니켈 수소	- 니켈 카드뮴보다 용량이 1.8배 가량 크다. - 전류 용량의 3배가 되는 대전류를 방전할 수 있다. - 니켈 카드뮴에서 니켈 수소로 바뀌어가고 있다.
	리튬 2차 전지	- 전압이 3.6 V로 높다. (니켈 카드뮴, 니켈 수소 전지 3개 직렬분과 같다.) - 고에너지 밀도이다. - 소형 경량이 중요시되는 휴대폰 또는 퍼스널 컴퓨터 등 최첨단 기기에 쓰인다. - 국내에는 없고 일본이나 미국에서 구해야 한다.

가면서 일반 만능기관으로 만들기는 어려워지고 있다. PCB를 뜨게 되면 칩을 DIP 타입이 아닌 SMD 타입으로 사용할 수가 있기 때문에 부피를 상당히 줄일 수 있다.

PCB를 뜨기 위해서는 artwork를 직접 해야 한다. Artwork라 하면 모든 패턴, 패드 등을 직접 그리는 작업을 말한다. PCB는 artwork를 한 그대로 나오게 되는 것이다. 물론 회로도만 완성하여 기업체에 artwork를 맡기면 모두 해 주지만 값이 상당히 비싸기 때문에 조금이라도 아끼기 위해서는 artwork까지는 본인이 직접 하는 것이 나올 것이다.

PCB 기관은 일반적으로 양면을 사용한다. 부품면의 COPPER, 부품면의 MASK, 부품면의 SILK, 납땀면의 COPPER, 납땀

면의 MASK, 납땀면의 SILK, DRILL 이렇게 총 7개의 필름을 제작하여 PCB를 뜨게 되는 것인데, 우리는 이 7개의 필름을 제작할 수 있도록 원본이 될 수 있는 7개의 그림을 그려 거버 파일 형태로 업체에 넘기면 되는 것이다.

부품면은 앞면을, 납땀면은 뒷면을 이야기하는 것이다. COPPER는 도선과 같은 전류가 통하는 곳을 말하고 MASK는 납이 붙어야 하는 곳을 말한다. DRILL은 구멍이 뚫려야 하는 곳을 말하고 마지막으로 SILK는 기관에 쓸 글씨와 그림 등을 말한다.

비전 시스템

비전 시스템은 로봇의 눈에 해당하는 역

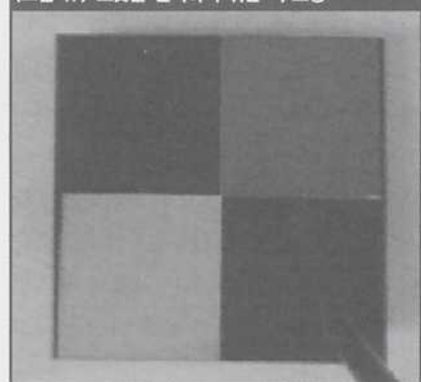
할을 하게 된다. 로봇은 눈이 없기 때문에 공이 현재 어디에 있는지를 알 수 있는 방법이 없다. 또한 자신의 정확한 위치도 모르기 때문에 비전의 역할은 상당히 중요하다.

비전 시스템을 구현하는데 있어서도 크게 두 가지로 나누어 생각할 수 있을 것이다. 하나는 하드웨어적으로 화상을 분석하여 원하는 물체의 좌표 등을 알아내는 방법이고, 또 하나는 하드웨어로는 단지 화상을 캡처하는 데에만 사용하고 나머지는 모두 소프트웨어적으로 처리하는 방법이다. 당연히 하드웨어를 사용하여 좌표를 알아내는 방법이 빠르고 안정되지만 화상처리용 보드는 상당히 고가이고 또한 직접 제작한다고 해도 어려움이 많을 것이다. 필자가 속한 팀에서도 소프트웨어를 이용하여 화상을 분석하는 방식을 채택하였다.

MIRAGE I 로봇의 가장 윗부분에는 그림 17과 같은 patch를 붙였다. 아래의 노란색의 사각형은 팀의 색깔을 나타내는 것이고, 위의 보라색 사각형은 로봇의 ID를 나타내는 색을 나타내는 것이다.

비전 시스템에서는 캡처 보드를 이용하여 화상을 캡처하여 먼저 RAM Drive에 화상을 파일로 저장한 후 다시 파일을 열어 우리가 원하는 색깔을 찾아내는 방식으로 프로 그래밍 하였다. 또한 팀의 색깔과 로봇 ID를 나타내는 색깔을 모두 찾은 후에는 팀 색의 중심과 로봇 ID 색의 중심을 연결하여 벡터를 구하고 그것을 이용해 현재 로봇이 바라 보고 있는 방향까지 구하도록 하였다.

〈그림 17〉 로봇을 인식하기 위한 색 도형



비전 시스템에서 가장 문제가 되는 것은 외부의 조명에 따라 화상이 너무 크게 변한다는 것이다. 따라서 장소가 바뀌게 되거나 조명이 바뀌게 되면 색상에 변화가 생기게 되어 프로그램 상에서 세팅을 다시 해주어야 한다.

또한 노이즈도 상당히 섞여 있어 노이즈를 제거해주는 알고리즘도 포함시켜야 한다. 참고로 한 가지 충고를 하고 싶은 것은 비전용 카메라는 고가여도 좋은 것을 구입해야 한다는 것이다.

필자 팀의 경우 비전용 카메라는 값싼 것으로 구입했더니 색상의 구별에도 어려움이 생기게 되고 노이즈도 상당히 많이 섞여 프로그래밍 하는데 어려움이 많았다. 또한 비전 시스템은 빨라야 한다. 본 팀의 시스템은 1초에 약 2frame 정도를 분석하여 로봇은 항상 공의 0.5초 전의 좌표만을 전송 받게 되어 공을 뒤따라가기만 하는 결과를 보였던 96년의 경우와는 달리, 97년에는 1초에 30 frame 정도를 분석하여 비전은 별 문제가 되지 않았다. 다만 색깔이 조명에 따라 영향을 많이 받는다는 단점이 있었다.

알고리즘

필자가 생각하기에는 로봇의 하드웨어를 제작하는 시간보다 더 많은 시간을 투자해야 하는 부분이 바로 알고리즘이다.

여러 대의 로봇과 호스트 컴퓨터로 한 팀을 이루어 경기를 진행해야 하기 때문에 알고리즘은 무궁무진한 방향으로 개발할 수 있을 것이다. 다시 말하면 최적의 알고리즘이란 있을 수 없다는 뜻이다.

로봇의 하드웨어에 맞는 알고리즘을 개발해야 할 것이고 비전 시스템의 성능, 통신 속도, 로봇의 처리속도, 로봇의 움직임의 최대 속도 등의 요소들이 모두 고려되어야 할 것이다. 다음에 설명하는 필자의 알고리즘이 가장 좋은 방법이라고는 말할 수 없지만 처음 시작하는 사람들을 위해 도움을 주고자 언급하겠다.

로봇은 공격수, 수비수, 골키퍼 3대로 구성되는데 각각의 경우 모두를 프로그래밍해야 할 것이다. 필자는 로봇마다 공격수, 수비수, 골키퍼를 부여하지 않았다. 모든 로봇은 같은 프로그램을 사용하도록 했고 프로그램 상에서 상황에 따라 자신이 공격수의 역할을 할 것인지, 수비수의 역할을 할 것인지, 아니면 골키퍼의 역할을 할 것인지를 결정하도록 하였다. 즉 하나의 프로그램에 세 가지의 역할이 모두 포함되어 있는 것이다.

우선 공격수는 공이 중앙선을 넘어 상대 진영으로 들어가게 되면 본격적으로 공격을 시작한다. 먼저 공으로부터 뒤로 약 10cm 떨어진 위치까지 간다. 이때 그냥 뒤로 10cm가 아니고 상대팀의 골대와 공의 좌표 사이의 벡터를 구하여 그 벡터의 반대 방향으로 10cm 떨어진 곳으로 가는 것이다. 그 위치에 도착한 공격수는 골대의 방향을 바라보기 위해 다시 공으로부터 골대로의 벡터 방향으로 약 5cm 정도 움직인다.

이렇게 되면 로봇은 공을 앞에 두고 상대팀의 골대를 바라보게 된다. 만약을 위해 자신의 센서를 가지고 정말 자신의 전방에 공이 있는지를 스캐닝 하게 된다. 공을 발견하게 되면 그 즉시 강한 슛을 날리게 되는 것이다.

이때 슛의 종류로는 세 가지를 구현했다. 한 가지는 빠른 속도로 전진하여 공을 미는 것이고 또 한 가지는 약 5cm 정도 후진하였다가 빠른 속도로 공을 향해 달려가 치는 것이며, 마지막 한 가지는 만약 공을 찾긴 했는데 자신이 현재 보고 있는 방향이 상대팀 골대가 아니고 우리팀 골대인 경우 공의 옆으로 달려가 상대팀 골대 쪽으로 회전을 하면서 터닝슛을 하는 것이다.

수비수는 공이 중앙선을 기준으로 우리팀 진영에 있을 경우 공의 앞으로 달려가 막고 상황에 따라 공격수의 기능과 같이 상대 골대를 향하여 룡슛을 날리게 된다. 마지막으로 골키퍼는 경기장을 중앙선과 수직으로 구역을 분할하여 공이 현재 있는 구역에 따

라 움직이게 된다.

또한 만약 수비지역에 공이 오랜 시간동안 머물러 있게 되면 골키퍼는 달려가 수비지역 밖으로 공을 차내게 된다. 공격수의 기능과 같이 공을 차낸 후에는 다시 공이 있는 구역에 해당하는 골대 앞으로 가서 골대를 지키고 있게 된다. 물론 모든 로봇들은 장애물 센서를 가지고 있어 움직일 때 다른 팀의 로봇 등을 피하게 된다. 하지만 때에 따라서는 장애물이 있어도 공을 차기 위해 밀어붙이는 경우도 있다. 이것은 전략적으로 프로그래밍 한 것이다. 로봇 프로그램의 알고리즘 흐름도를 간단히 나타내보면 그림 18과 같다.

통신은 호스트 컴퓨터에서 데이터를 보내면 로봇 마이크로컨트롤러 보드에서는 받기 인터럽트가 걸리게 되어 받기 인터럽트 함수를 부르게 된다. 받기 인터럽트 함수에서는 먼저 통신 데이터의 op-code를 분석한 후 op-code에 따라 뒤에 날아오는 데이터를 구분하게 된다.

그리고 통신 시 error가 발생하는 것을 막기 위해서 맨체스터 코딩을 하여 호스트에서 송신을 하고 로봇에서는 다시 디코딩을 하도록 프로그래밍 하였다. 호스트 컴퓨터에서는 로봇이 움직이기 시작하도록 하는 시작 명령, 모두 정지하라는 리셋 명령, 공격 방향의 설정, 공격수와 수비수, 골키퍼의 교대 등의 신호를 보낼 수 있다.

장애물 센싱은 100ms~300ms마다 일정하게 프로그램적인 인터럽트를 발생시키고 5초의 센서를 이용하여 주변의 장애물의 여부를 판단한다. 만약 전방에 장애물이 감지될 경우에는 그 즉시 충돌하지 않도록 방향을 약간 변환한다. 단 슛을 하려고 하는 순간에는 장애물을 피하면 우리 팀에게 손해이므로 그 때에는 강제로 프로그램 타이머 인터럽트를 disable시키게 된다.

간단히 필자가 사용했던 알고리즘에 대해 설명해 보았다. 실제로 위에서 설명한 알고리즘을 어떻게 구현하였는가에 대해 궁금하게 여기는 사람들을 위해 97년 대회에 사용

했던 로봇용 프로그램 source를 첨부하였다. C로 프로그래밍 하였고 ic96 컴파일러를 사용하였다.

결론

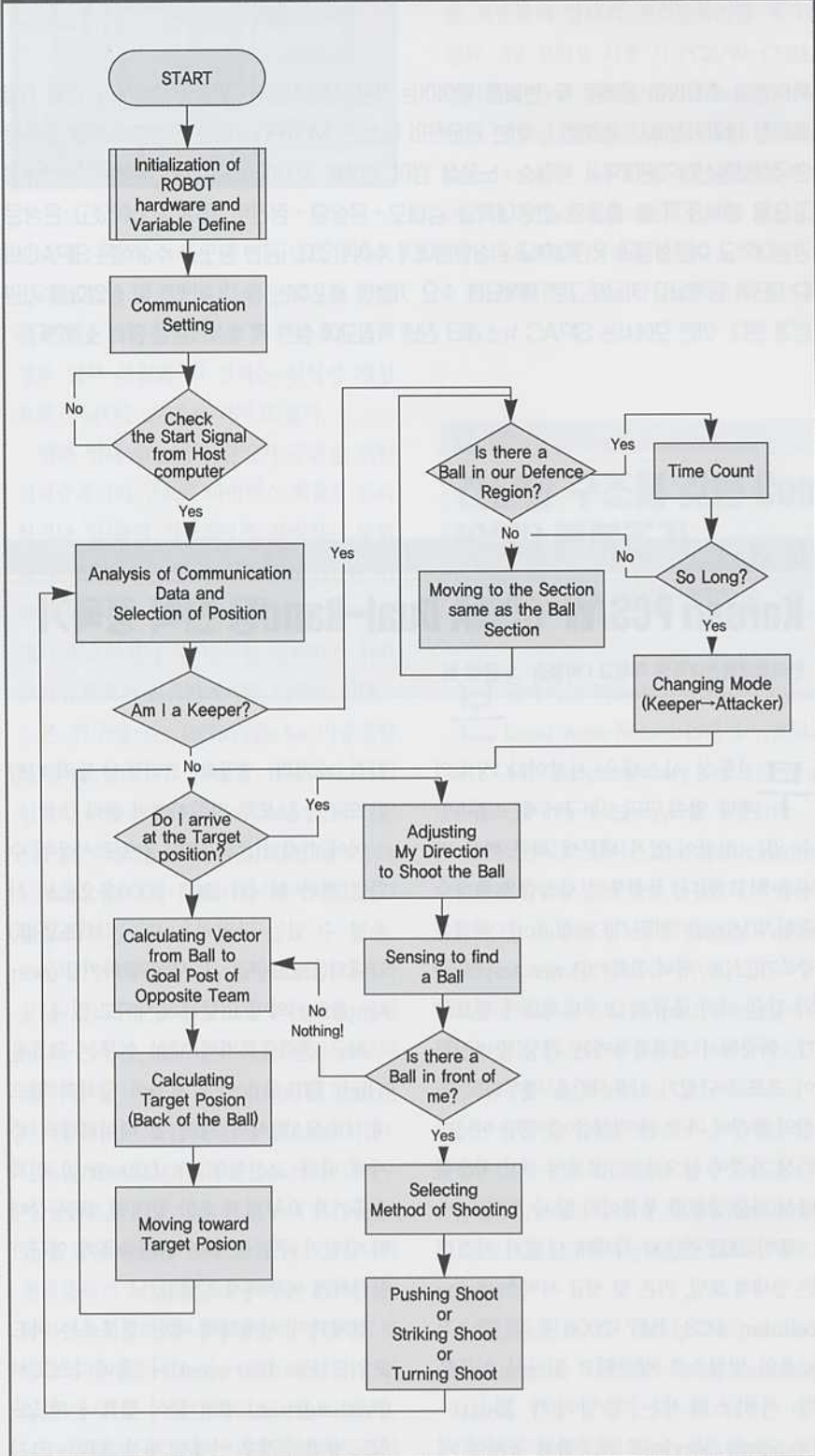
로봇 축구 시스템은 마이크로컨트롤러,

센서, 제어, 비전 시스템, 통신, 전력 시스템 등 대부분의 하드웨어 영역을 포함하고 영상 알고리즘, 다개체 로봇 시스템을 위한 알고리즘 등 소프트웨어 영역 등을 포함하는 등 여러 분야 기술의 집합체이다.

현재 대부분의 팀들이 로봇과 비전 시스템 등의 하드웨어에 의한 문제는 거의 해결한 상태이다. 이제는 전략 알고리즘과 다개체 시스템을 얼마나 잘 구현하는가 하는 것이 그 팀의 우수성을 입증할 수 있는 유일한 길이 될 것이다.

마지막으로 최근의 로봇 축구 팀들은 모두 비전 중심 구조로 시스템을 구현하고 있다. 따라서 로봇의 하드웨어는 빠르고 힘센 모터와 간단한 마이크로컨트롤러 보드, 그리고 RF 통신 모듈만으로 구현된다. 이제까지 설명한 MIRAGE I 로봇은 최근의 로봇 축구 시스템 구조와는 다르지만 센서 회로, 통신 회로, 로봇 프로그램 등은 다른 로봇을 구현하는데 많은 도움이 되리라 생각된다.

<그림 18> 로봇 알고리즘 흐름도



MIRAGE 소개

MIRAGE는 Micro Robot AGgrEgation의 약자로 로봇 축구 대회를 위해 만들었던 모임이 모태가 되어 발전한 동아리이다. MIRAGE는 신기루라는 뜻으로 과거에 환상으로만 생각했던 것들을 실현하겠다는 의지를 내포하고 있다. KAIST 전기 및 전자공학과 내의 동아리로 축구 로봇뿐만 아니라 로봇과 관련된 모든 영역을 다루고 있으며, 현재 여러 가지 프로젝트들을 수행중에 있다. **511**