





Code convention & Rule

코드 업로드 및 관리 규칙

1. GitHub로 개발 중인 코드를 관리한다.
 - a. Branch를 이용하여 동시에 개발 작업을 수행한다.
 - b. commit message는 최근 작업 내용을 요약하여 작성한다.
 - c. 각 branch를 merge할 때 reviewer를 선정하고, review에서 문제가 없을 경우 merge를 진행한다.
 - d. 최종적으로 업데이트된 코드는 main branch에 업로드한다.
2. 각 모듈의 소스코드, 헤더 파일은 묶어서 파일로 만든다.
 - a. Main Code는 밖에 따로 빼두기
 - b. ex)  BSW --  ASCLIN--  Asclin.c /  Asclin.h

코딩 규칙

1. 폴더 이름은 무조건 대문자로만 작성한다.
2. 하위 파일들은 첫 글자만 대문자로 작성하고 이하 글자는 소문자로 작성한다.
3. 함수 명명 규칙 : 단어 첫 글자는 대문자를 사용하며 이하 글자는 소문자로 작성한다. 띄어쓰기는 _로 구분하며 띄어쓰기를 한 경우 새로운 단어를 작성해야 한다. ex) System_Init();
 - a. 함수 이름 앞에는 파일 명을 반드시 붙여야 한다. ex) Can_Get_Dist();
 - b. 예외 : Asclin.c 내부 함수들은 명명 규칙에서 제외한다.
 - i. void Asclin0_InitUart(void);
void Asclin0RxIsrHandler(void);

int Asclin0_PollUart(unsigned char *chr);
unsigned char Asclin0_InUart(void);
unsigned char Asclin0_InUartNonBlock(void);
void Asclin0_OutUart(const unsigned char chr);

void remove_null(char *s);
void my_puts(const char *str);
void my_printf(const char *fmt, ...);
void my_scanf(const char *fmt, ...);
 - c. 예외 : Led.c 내부 delay_ms(unsigned int ms): 명명 규칙에서 제외한다.