Green sheet ISA Pipeline CPU

	System			
	Id			Null
	Type	Op	Null	
	31:30	29:28	27:26	25:0
NOP	00	00	00	00000000000000000000000000
COM	00	01	00	00000000000000000000000000
END	00	10	00	00000000000000000000000000

	Data register-register						
	Id			Registers			Null
	Type	Op	I	RD	RA	RB	
	31:30	29:27	26	25:22	21:18	17:14	13:0
ADD	01	000	0	xxxx	xxxx	xxxx	00000000000000
SUB	01	001	0	xxxx	xxxx	xxxx	00000000000000
MOV	01	010	0	xxxx	0000	xxxx	00000000000000
MUL	01	011	0	xxxx	xxxx	xxxx	00000000000000
CMP	01	100	0	0000	xxxx	xxxx	00000000000000

	Data register-immediate					
	Id			Registers		
	Tipo	Op	I	RD	RA	Immediate
	31:30	29:27	26	25:22	21:18	17:0
ADD	01	000	1	xxxx	xxxx	xxxxxxxxxxxxxxxxxx
SUB	01	001	1	xxxx	xxxx	xxxxxxxxxxxxxxxxxx
MOV	01	010	1	xxxx	0000	xxxxxxxxxxxxxxxxxx
MUL	01	011	1	xxxx	xxxx	xxxxxxxxxxxxxxxxxx
CMP	01	100	1	0000	xxxx	xxxxxxxxxxxxxxxxxx

	Memoria					
	Id			Addressing		
	Type	Op	Null	RD	RA	Immediate
	31:30	29	28:26	25:22	21:18	17:0
LDR	10	0	000	xxxx	xxxx	xxxxxxxxxxxxxxxxxx
STR	10	1	000	xxxx	xxxx	xxxxxxxxxxxxxxxxxx

	Control				
	Id			Null	Instruction number
	Type	Op	Null		Immediate
	31:30	29:28	27:26	25:18	17:0
JMP	11	00	00	00000000	xxxxxxxxxxxxxxxxxx
JEQ	11	01	00	00000000	xxxxxxxxxxxxxxxxxx
JLT	11	10	00	00000000	xxxxxxxxxxxxxxxxxx

INSTRUCCIÓN	MODO	EJEMPLO	SINTAXIS	COMPORTAMIENTO	OP CODE	TIPO
NOP	no aplica	NOP	NOP	PC ← PC + 4; Flags ← 0	0	system
COM	no aplica	COM	COM	COMFlag ← 1	1	
END	no aplica	END	END	ENDFlag ← 1	2	
ADD	registro	ADD R3, R4, R5	ADD Rd, Ra, Rb	Rd ← Ra + Rb	0	dato
ADD	inmediato	ADD R3, R4, #5	ADD Rd, Ra, Imm	Rd ← Ra + Ext. Imm		
SUB	registro	SUB R9, R8, R3	SUB Rd, Ra, Rb	Rd ← Ra – Rb	1	
SUB	inmediato	SUB R9, R8, #2	SUB Rd, Ra, Imm	Rd ← Ra – Ext. Imm		
MOV	registro	MOV R4, R5	MOV Rd, Rb	Rd ← Rb	2	
MOV	inmediato	MOV R4, #20	MOV Rd, Imm	Rd ← Ext. Imm		
MUL	registro	MUL R1, R0, R2	MUL Rd, Ra, Rb	Rd ← Ra • Rb	3	
MUL	inmediato	MUL R1, R0, #7	MUL Rd, Ra, Imm	Rd ← Ra • Ext. Imm		
CMP	registro	CMP R6, R7	CMP Ra, Rb	Rn – Rm; ALUFLags ← NZ	4	
CMP	inmediato	CMP R6, #8	CMP Ra, Imm	Rn – Ext. Imm; ALUFLags ← NZ		
LDR	registro	LDR R4, [R5]	LDR Rd, [Ra]	Rd ← MEM[Ra + 0]	0	memoria
LDR	inmediato	LDR R4, [R5, #8]	LDR Rd, [Ra, Imm]	Rd ← MEM[Ra + Ext. Imm]		
STR	registro	STR R1, [R6]	STR Rd, [Ra]	MEM[Ra + 0] ← Rd	1	
STR	inmediato	STR R1, [R6, #52]	STR Rd, [Ra, Imm]	MEM[Ra + Ext. Imm] ← Rd		
JMP	incondicional	JMP _loop	JMP LABEL	PC ← Ext. Imm	0	control
JEQ	condicional	JEQ _loop	JEQ LABEL	PC ← Ext. Imm if Z=1 else PC+4	1	
JLT	condicional	JLT _loop	JLT LABEL	PC ← Ext. Imm if N=1 else PC+4	2	

INSTRUCCIÓN	NOMBRE	DESCRIPCIÓN/PROPÓSITO
NOP	*No Operation*	Esta instrucción no hace nada durante la ejecución. Se utiliza con fines de temporización, para prevenir riesgos y para invalidar una instrucción existente, como un salto, como objetivo de una instrucción de ejecución.
COM	*Communication*	Esta instrucción activa la bandera de COM. Se utiliza como indicador para iniciar la comunicación con el arduino.
END	*End*	Esta instrucción activa la bandera de END. Se utiliza como indicador para indicar la finalización del algoritmo.
ADD	*Add*	Esta instrucción realiza la suma de dos valores y los almacena en un registro destino.
SUB	*Substract*	Esta instrucción realiza la resta de dos valores y los almacena en un registro destino.
MOV	*Move*	Esta instrucción almacena el valor de un registro o inmediato recibido en un registro destino.
MUL	*Multiply*	Esta instrucción realiza la multiplicación de dos valores y los almacena en un registro destino.
CMP	*Compare*	Esta instrucción realiza la resta de dos valores y actualiza las banderas de la ALU acorde al resultado que obtiene.
LDR	*Load*	Esta instrucción guarda en el primer registro de la instrucción el valor ubicado en la dirección de memoria efectiva conformada por el valor del segundo registro adicionado a un indice que puede ser indicado con un inmediato u otro registro. Si no se indica un indice, se asume un cero y la dirección efectiva es igual al segundo registro.
STR	*Store*	Esta instrucción guarda el valor del primer registro de la instrucción en la dirección de memoria efectiva conformada por el valor del segundo registro adicionado a un indice que puede ser indicado con un inmediato u otro registro. Si no se indica un indice, se asume un cero y la dirección efectiva es igual al segundo registro.
JMP	*Jump*	Esta instrucción realiza un salto a la primera instrucción después de la etiqueta que se indica.
JEQ	*Jump equal*	Esta instrucción comprueba si la bandera Z (cero) se ha accionado, si es así realiza un salto a la primera instrucción después de la etiqueta correspondiente, sino continua con la siguiente instrucción. **NOTA:** La instrucción CMP debe ser utilizada antes de emplear esta instrucción, en caso contrario podría tener un funcionamiento incierto.
JLT	*Jump less than*	Esta instrucción comprueba si la bandera N (negativo) se ha accionado, si es así realiza un salto a la primera instrucción después de la etiqueta correspondiente, sino continua con la siguiente instrucción. **NOTA:** La instrucción CMP debe ser utilizada antes de emplear esta instrucción, en caso contrario podría tener un funcionamiento incierto.

REGISTROS	DESCRIPCIÓN/PROPÓSITO
R0 - R9	Se emplean diez registros de propósito general.