

Digital Input	ModbusTCP Address		Python sample code to access input
I1	1	Bit / Coil	lg8.read_discrete_inputs(lg8add(1),
I2	2	Bit / Coil	lg8.read_discrete_inputs(lg8add(2), 1)
I3	3	Bit / Coil	lg8.read_discrete_inputs(lg8add(3), 1)
I4	4	Bit / Coil	lg8.read_discrete_inputs(lg8add(4), 1)
I5	5	Bit / Coil	lg8.read_discrete_inputs(lg8add(5), 1)
I6	6	Bit / Coil	lg8.read_discrete_inputs(lg8add(6), 1)
I7	7	Bit / Coil	lg8.read_discrete_inputs(lg8add(7), 1)
I8	8	Bit / Coil	lg8.read_discrete_inputs(lg8add(8), 1)
I9	9	Bit / Coil	...
I10	10	Bit / Coil	...
I11	11	Bit / Coil	...
I12	12	Bit / Coil	...
I13	13	Bit / Coil	...
I14	14	Bit / Coil	...
I15	15	Bit / Coil	...
I16	16	Bit / Coil	...
I17	17	Bit / Coil	...
I18	18	Bit / Coil	...
I19	19	Bit / Coil	...
I20	20	Bit / Coil	...
I21	21	Bit / Coil	...
I22	22	Bit / Coil	...
I23	23	Bit / Coil	...
I24	24	Bit / Coil	...

Logo +1 Offset
korrigieren

Logo!8 Modbus TCP
Bit/Register Adresse

Anzahl der zu lesenden
Bit/Byte

Spalte 3 setzt voraus, dass die Python ModbusTCP Bibliothek installiert wurde:

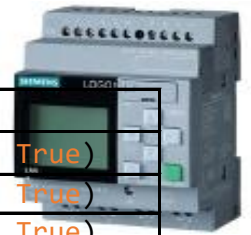
```
$ pip install pymodbus
```

```
def lg8add(logo_modbus_tcp_address: int) -> int:
    """
    Logo8 uses modbusTCP +1 address offset
    to stay conform with their documentation, subtract 1 from each given address
    """
    return logo_modbus_tcp_address-1
```

Address Type	Range	Mapped Modbus Address	Direction	Unit
I	1 - 24	Discrete Input (DI) 1 - 24	R	bit
Q	1 - 20	Coil 8193 - 8212	R/W	bit
M	1 - 64	Coil 8257 - 8320	R/W	bit
V	0.0 - 850.7	Coil 1 - 6808	R/W	bit
AI	1 - 8	Input Register (IR) 1 - 8	R	word
VW	0 - 850	Holding Register (HR) 1 - 425	R/W	word
AQ	1 - 8	Holding Register (HR) 513 - 520	R/W	word
AM	1 - 64	Holding Register (HR) 529 - 592	R/W	word

Discrete Input





Digital Output	ModbusTCP Address		Python sample code to access output
DO1	8193	Bit / Coil	<code>lg8.write_single_coil(lg8add(8193), True)</code>
DO2	8194	Bit / Coil	<code>lg8.write_single_coil(lg8add(8194), True)</code>
DO3	8195	Bit / Coil	<code>lg8.write_single_coil(lg8add(8195), True)</code>
DO4	8196	Bit / Coil	<code>lg8.write_single_coil(lg8add(8196), True)</code>
DO5	8197	Bit / Coil	...
DO6	8198	Bit / Coil	...
DO7	8199	Bit / Coil	...
DO8	8200	Bit / Coil	...
DO9	8201	Bit / Coil	...
DO10	8202	Bit / Coil	...
DO11	8203	Bit / Coil	...
DO12	8204	Bit / Coil	...
DO13	8205	Bit / Coil	...
DO14	8206	Bit / Coil	...
DO15	8207	Bit / Coil	...
DO16	8208	Bit / Coil	...
DO17	8209	Bit / Coil	...
DO18	8210	Bit / Coil	...
DO19	8211	Bit / Coil	...
DO20	8212	Bit / Coil	...

in das Coil/Register zu schreibender Wert

Auf die digitalen Ausgänge kann auch lesend zugegriffen werden, um z.B. den Status abzufragen:

```
do1_r = lg8.read_coils(lg8add(8193), 1)
```

Address Type	Range	Mapped Modbus Address	Direction	Unit
I	1 - 24	Discrete Input (DI) 1 - 24	R	bit
Q	1 - 20	Coil 8193 - 8212	R/W	bit
M	1 - 64	Coil 8257 - 8320	R/W	bit
V	0.0 - 850.7	Coil 1 - 6808	R/W	bit
AI	1 - 8	Input Register (IR) 1 - 8	R	word
VW	0 - 850	Holding Register (HR) 1 - 425	R/W	word
AQ	1 - 8	Holding Register (HR) 513 - 520	R/W	word
AM	1 - 64	Holding Register (HR) 529 - 592	R/W	word

Coil





Memory (Flags)	ModbusTCP Address		Python sample code to access coil
M1	8257	Bit / Coil	...
M2	8258	Bit / Coil	...
M3	8259	Bit / Coil	...
M4	8260	Bit / Coil	...
M5	8261	Bit / Coil	...
M6	8262	Bit / Coil	...
M7	8263	Bit / Coil	...
M8	8264	Bit / Coil	...
M9	8265	Bit / Coil	...
M10	8266	Bit / Coil	...
M11	8267	Bit / Coil	...
M12	8268	Bit / Coil	...
M13	8269	Bit / Coil	...
M14	8270	Bit / Coil	...
M15	8271	Bit / Coil	...
M16	8272	Bit / Coil	...
M17	8273	Bit / Coil	...
M18	8274	Bit / Coil	...
M19	8275	Bit / Coil	...
M20	8276	Bit / Coil	...
M21	8277	Bit / Coil	...
M22	8278	Bit / Coil	...
M23	8279	Bit / Coil	...
M24	8280	Bit / Coil	...
M25	8281	Bit / Coil	...
M26	8282	Bit / Coil	<code>lg8.write_single_coil(lg8add(8282), True)</code>
M27	8283	Bit / Coil	...
M28	8284	Bit / Coil	...
M29	8285	Bit / Coil	...
M30	8286	Bit / Coil	<code>lg8.write_single_coil(lg8add(8286), True)</code>
M31	8287	Bit / Coil	<code>lg8.write_single_coil(lg8add(8287), True)</code>
M32	8288	Bit / Coil	...
M33	8289	Bit / Coil	...
M34	8290	Bit / Coil	...
M35	8291	Bit / Coil	...
M36	8292	Bit / Coil	...
M37	8293	Bit / Coil	...
M38	8294	Bit / Coil	...
M39	8295	Bit / Coil	...
M40	8296	Bit / Coil	...
M41	8297	Bit / Coil	...
M42	8298	Bit / Coil	...
M43	8299	Bit / Coil	...
M44	8300	Bit / Coil	...
M45	8301	Bit / Coil	...





M46	8302	Bit / Coil	...
M47	8303	Bit / Coil	...
M48	8304	Bit / Coil	...
M49	8305	Bit / Coil	...
M50	8306	Bit / Coil	...
M51	8307	Bit / Coil	...
M52	8308	Bit / Coil	...
M53	8309	Bit / Coil	...
M54	8310	Bit / Coil	...
M55	8311	Bit / Coil	...
M56	8312	Bit / Coil	...
M57	8313	Bit / Coil	...
M58	8314	Bit / Coil	...
M59	8315	Bit / Coil	...
M60	8316	Bit / Coil	...
M61	8317	Bit / Coil	...
M62	8318	Bit / Coil	...
M63	8319	Bit / Coil	...
M64	8320	Bit / Coil	...

Auf die digitalen Merker kann lesend und schreibend zugegriffen werden:

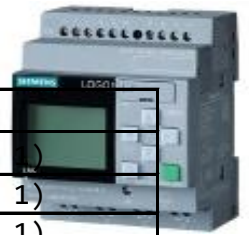
```
m26 = lg8.read_coils(lg8add(8282), 1)
```

Modbus address space

Address Type	Range	Mapped Modbus Address	Direction	Unit
I	1 - 24	Discrete Input (DI) 1 - 24	R	bit
Q	1 - 20	Coil 8193 - 8212	R/W	bit
M	1 - 64	Coil 8257 - 8320	R/W	bit
V	0.0 - 850.7	Coil 1 - 6808	R/W	bit
AI	1 - 8	Input Register (IR) 1 - 8	R	word
VW	0 - 850	Holding Register (HR) 1 - 425	R/W	word
AQ	1 - 8	Holding Register (HR) 513 - 520	R/W	word
AM	1 - 64	Holding Register (HR) 529 - 592	R/W	word

Coil





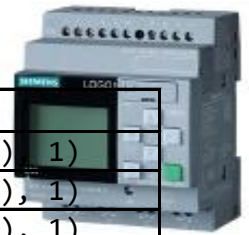
Analog Inputs	ModbusTCP Address		Python sample code to access register
AI1	1	Register	<code>lg8.read_input_registers(lg8add(1), 1)</code>
AI2	2	Register	<code>lg8.read_input_registers(lg8add(2), 1)</code>
AI3	3	Register	<code>lg8.read_input_registers(lg8add(3), 1)</code>
AI4	4	Register	<code>lg8.read_input_registers(lg8add(4), 1)</code>
AI5	5	Register	<code>lg8.read_input_registers(lg8add(5), 1)</code>
AI6	6	Register	<code>lg8.read_input_registers(lg8add(6), 1)</code>
AI7	7	Register	<code>lg8.read_input_registers(lg8add(7), 1)</code>
AI8	8	Register	<code>lg8.read_input_registers(lg8add(8), 1)</code>

ModbusTCP sieht bei Registern (z.B. Input Register, Holding Register) immer 16 Bit vor (2 Byte, Word)

Modbus address space				
Address Type	Range	Mapped Modbus Address	Direction	Unit
I	1 - 24	Discrete Input (DI) 1 - 24	R	bit
Q	1 - 20	Coil 8193 - 8212	R/W	bit
M	1 - 64	Coil 8257 - 8320	R/W	bit
V	0.0 - 850.7	Coil 1 - 6808	R/W	bit
AI	1 - 8	Input Register (IR) 1 - 8	R	word
VW	0 - 850	Holding Register (HR) 1 - 425	R/W	word
AQ	1 - 8	Holding Register (HR) 513 - 520	R/W	word
AM	1 - 64	Holding Register (HR) 529 - 592	R/W	word

Input Register





VW	ModbusTCP Address		Python sample code to access register
VW0	1	Register	lg8.read_holding_registers(lg8add(1), 1)
VW2	2	Register	lg8.read_holding_registers(lg8add(2), 1)
VW4	3	Register	lg8.read_holding_registers(lg8add(3), 1)
VW6	4	Register	lg8.read_holding_registers(lg8add(4), 1)
VW8	5	Register	...
VW10	6	Register	...
VW12	7	Register	...
VW14	8	Register	...
VW16	9	Register	...
VW18	10	Register	...
VW20	11	Register	...
VW22	12	Register	...
VW24	13	Register	...
VW26	14	Register	...
VW28	15	Register	lg8.write_single_register(lg8add(15), 800)
VW30	16	Register	lg8.write_single_register(lg8add(16), 1200)
...	...	Register	...
...	...	Register	...
VW846	424	Register	lg8.read_holding_registers(lg8add(424), 1)
VW848	425	Register	lg8.read_holding_registers(lg8add(425), 1)

Die VM Bereiche in der Logo sind quasi spezielle Speicherbereiche zur Kommunikation mit anderen Devices im gleichen Netzwerk. Konfiguration über *Tools --> Parameter VM Mapping...*

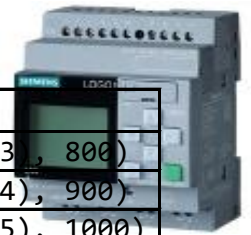
Auf die VM Bereiche kann lesend und schreibend zugegriffen werden.

Address Type	Range	Mapped Modbus Address	Direction	Unit
I	1 - 24	Discrete Input (DI) 1 - 24	R	bit
Q	1 - 20	Coil 8193 - 8212	R/W	bit
M	1 - 64	Coil 8257 - 8320	R/W	bit
V	0.0 - 850.7	Coil 1 - 6808	R/W	bit
AI	1 - 8	Input Register (IR) 1 - 8	R	word
VW	0 - 850	Holding Register (HR) 1 - 425	R/W	word
AQ	1 - 8	Holding Register (HR) 513 - 520	R/W	word
AM	1 - 64	Holding Register (HR) 529 - 592	R/W	word

Holding Register

ID	Block	Parameter	Type	Address
1	B001 AnInAmp_1 [Analog Amplifier]	Ax, amplified	Word	0
2	B002 AnInAmp_2 [Analog Amplifier]	Ax, amplified	Word	2
3	B003 AnInAmp_3 [Analog Amplifier]	Ax, amplified	Word	4
4	B004 AnInAmp_4 [Analog Amplifier]	Ax, amplified	Word	6
5				



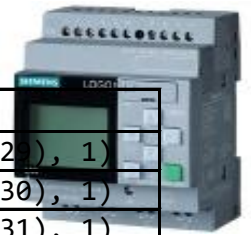


Analog Outputs	ModbusTCP Address		Python sample code to access register
AO1	513	Register	<code>lg8.write_single_register(lg8add(513), 800)</code>
AO2	514	Register	<code>lg8.write_single_register(lg8add(514), 900)</code>
AO3	515	Register	<code>lg8.write_single_register(lg8add(515), 1000)</code>
AO4	516	Register	<code>lg8.write_single_register(lg8add(516), 1100)</code>
AO5	517	Register	<code>lg8.write_single_register(lg8add(517), 1200)</code>
AO6	518	Register	...
AO7	519	Register	...
AO8	520	Register	...

Address Type	Range	Mapped Modbus Address	Direction	Unit
I	1 - 24	Discrete Input (DI) 1 - 24	R	bit
Q	1 - 20	Coil 8193 - 8212	R/W	bit
M	1 - 64	Coil 8257 - 8320	R/W	bit
V	0.0 - 850.7	Coil 1 - 6808	R/W	bit
AI	1 - 8	Input Register (IR) 1 - 8	R	word
VW	0 - 850	Holding Register (HR) 1 - 425	R/W	word
AQ	1 - 8	Holding Register (HR) 513 - 520	R/W	word
AM	1 - 64	Holding Register (HR) 529 - 592	R/W	word

Holding Register





Analog Memory	ModbusTCP Address		Python sample code to access register
1	529	Register	<code>lg8.read_holding_registers(lg8add(529), 1)</code>
2	530	Register	<code>lg8.read_holding_registers(lg8add(530), 1)</code>
3	531	Register	<code>lg8.read_holding_registers(lg8add(531), 1)</code>
4	532	Register	<code>lg8.read_holding_registers(lg8add(532), 1)</code>
5	533	Register	<code>lg8.read_holding_registers(lg8add(533), 1)</code>
6	534	Register	...
7	535	Register	...
8	536	Register	...
9	537	Register	...
10	538	Register	...
11	539	Register	...
12	540	Register	...
13	541	Register	...
14	542	Register	...
15	543	Register	...
16	544	Register	...
17	545	Register	...
18	546	Register	...
19	547	Register	...
20	548	Register	...
21	549	Register	...
22	550	Register	...
23	551	Register	...
24	552	Register	...
25	553	Register	...
26	554	Register	...
27	555	Register	...
28	556	Register	...
29	557	Register	...
30	558	Register	...
31	559	Register	...
32	560	Register	...
33	561	Register	...
34	562	Register	...
35	563	Register	...
36	564	Register	...
37	565	Register	...
38	566	Register	...
39	567	Register	...
40	568	Register	...
41	569	Register	...
42	570	Register	...
43	571	Register	...
44	572	Register	...
45	573	Register	...





46	574	Register	...
47	575	Register	...
48	576	Register	...
49	577	Register	...
50	578	Register	...
51	579	Register	...
52	580	Register	...
53	581	Register	...
54	582	Register	...
55	583	Register	...
56	584	Register	...
57	585	Register	...
58	586	Register	...
59	587	Register	...
60	588	Register	lg8.write_single_register(lg8add(588), 800)
61	589	Register	lg8.write_single_register(lg8add(589), 900)
62	590	Register	lg8.write_single_register(lg8add(590), 1000)
63	591	Register	lg8.write_single_register(lg8add(591), 1100)
64	592	Register	lg8.write_single_register(lg8add(592), 1200)

Auf die analogen Merker kann sowohl lesend als auch schreibend zugegriffen werden.

Address Type	Range	Mapped Modbus Address	Direction	Unit
I	1 - 24	Discrete Input (DI) 1 - 24	R	bit
Q	1 - 20	Coil 8193 - 8212	R/W	bit
M	1 - 64	Coil 8257 - 8320	R/W	bit
V	0.0 - 850.7	Coil 1 - 6808	R/W	bit
AI	1 - 8	Input Register (IR) 1 - 8	R	word
VW	0 - 850	Holding Register (HR) 1 - 425	R/W	word
AQ	1 - 8	Holding Register (HR) 513 - 520	R/W	word
AM	1 - 64	Holding Register (HR) 529 - 592	R/W	word

Holding Register

